

UNIVERSIDAD AUTÓNOMA DE MADRID

ESCUELA POLITÉCNICA SUPERIOR



Herramienta flexible para la creación y  
gestión de cuestionarios Web adaptativos:  
Aplicación al diseño de encuestas de  
satisfacción sobre servicios en línea del INE

**Carlos de la Cruz Fernández-Pacheco**

Julio de 2013



# Resumen

---

En este proyecto se presenta una herramienta software en línea que permite la creación flexible y gestión de cuestionarios Web para uso del Instituto Nacional de Estadística (INE).

La presentación de la herramienta va acompañada de un estudio comparativo de sistemas similares existentes en el mercado, de una descripción de las características principales y aportaciones novedosas de la herramienta desarrollada, y de una explicación general de cómo se representan y generan cuestionarios.

Todas las tecnologías utilizadas en el desarrollo de la herramienta, tanto del lado cliente como del lado del servidor, se explican para una mejor comprensión de la misma.

Así mismo, la arquitectura de la herramienta se muestra en primer lugar de modo esquemático para más adelante detallar cada uno de sus bloques, explicando sus funcionalidades individuales y la manera en la que se relacionan y comunican.



# Índice

---

<b>1. Introducción .....</b>	<b>1</b>
1.1. Motivación y definición del problema .....	1
1.2. Objetivos del proyecto .....	1
1.3. Descripción general de la herramienta desarrollada .....	2
<b>2. Análisis de herramientas de edición de cuestionarios Web.....</b>	<b>4</b>
2.1. Descripción de herramientas existentes .....	4
2.2. Limitaciones de las herramientas similares.....	6
2.3. Aportaciones de la herramienta desarrollada .....	7
<b>3. Tecnologías empleadas .....</b>	<b>8</b>
3.1. Tecnologías de la parte servidor.....	8
3.1.1. Java Servlets .....	8
3.1.2. Java Server Pages (JSP) y JSP Standard Tag Library (JSTL) .....	10
3.1.3. Apache Struts II.....	11
3.1.4. Server Side Includes (SSI).....	15
3.1.5. Comunicación cliente-servidor: HTTP Request y HTTP Response .....	15
3.2. Tecnologías de la parte cliente .....	16
3.2.1. JavaScript .....	16
3.2.2. JavaScript Object Notation (JSON) .....	17
3.2.3. Gestión de la sesión en cliente: cookies .....	17
<b>4. Arquitectura de la herramienta desarrollada .....</b>	<b>18</b>
4.1. Diagrama de arquitectura .....	18
4.2. Descripción de los módulos de la arquitectura.....	20
4.2.1. Acceso a bases de datos .....	20
4.2.2. Gestión de cuestionarios.....	21
4.2.3. Interfaz gráfica.....	27
<b>5. Gestión de cuestionarios.....</b>	<b>28</b>
5.1. Estructuras de datos .....	28
5.2. Flujo adaptativo de preguntas en un cuestionario.....	34
5.2.1. Navegación entre preguntas.....	34
5.2.2. Mantenimiento de respuestas.....	36
5.3. Búsqueda y selección de cuestionarios .....	36

5.3.1.	Uso de patrones.....	36
5.3.2.	Interpretación de peticiones de búsqueda.....	37
<b>6.</b>	<b>Conclusiones .....</b>	<b>37</b>
<b>Anexo I.</b>	<b>Ejemplos de encuestas generadas .....</b>	<b>38</b>
<b>Anexo II.</b>	<b>Soporte multi-idioma .....</b>	<b>41</b>
<b>Anexo III.</b>	<b>Accesibilidad .....</b>	<b>42</b>

# Glosario

---

<b>Apache Struts</b>	Framework desarrollado por Apache para el desarrollo de aplicaciones Web basadas en el paradigma MVC.
<b>Apache Tomcat</b>	Servidor de aplicaciones con tecnologías Java Servlets y JSP utilizado para el desarrollo de esta herramienta.
<b>DAO</b>	<i>Data Access Object</i> . Objeto encargado de realizar acceso a base de datos.
<b>ER</b>	Modelo Entidad-Relación utilizado para representar el esquema de base de datos.
<b>Framework</b>	Marco de trabajo definido para el desarrollo y/o la implementación de una herramienta.
<b>HTML</b>	<i>HyperText Markup Language</i> . Es el lenguaje utilizado para crear las páginas web de la herramienta.
<b>HTTP</b>	<i>Hypertext Transfer Protocol</i> . Es un protocolo de intercambio de información en la web entre cliente y servidor.
<b>JSON</b>	<i>Javascript Object Notation</i> . Formato ligero de intercambio de datos.
<b>JSP</b>	<i>Java Server Pages</i> . Tecnología para crear páginas Web dinámicas.
<b>JSTL</b>	<i>JSP Standard Tag Library</i> . Librería de etiquetas que se utilizan para implementar JSP.
<b>MVC</b>	Modelo-Vista-Controlador. Patrón de arquitectura software que comprende las 3 capas mencionadas.
<b>Servlet</b>	Programa Java que se ejecuta dentro de un servidor Web. Recibe/ responde desde/a clientes Web vía HTTP.
<b>SQL</b>	<i>Structured Query Language</i> . Lenguaje declarativo utilizado para realizar operaciones en la base de datos.
<b>URL</b>	<i>Uniform Resource Locator</i> (en español, Localizador de Recurso Uniforme) se refiere a la dirección única que identifica un recurso disponible en Internet.
<b>VO</b>	<i>Value Object</i> . Objeto que representa un elemento y que contiene uno o más valores.
<b>XML</b>	<i>Extensible Markup Language</i> . Lenguaje de etiquetado para el intercambio de datos.





# **1. Introducción**

## **1.1. Motivación y definición del problema**

El motivo de este proyecto nace de la necesidad por parte de la institución donde se ha realizado, el Instituto Nacional de Estadística (INE), de desarrollar un modelo que permita crear y administrar **cuestionarios Web** de manera sencilla, accesible y flexible a cambios, con los que cualquier persona trabajadora del INE pueda construir encuestas de satisfacción sobre servicios del sitio Web del INE.

Las encuestas de satisfacción son estudios que tienen como fin conocer el grado de satisfacción del usuario sobre un servicio a partir de unas preguntas que propone el administrador de dicho servicio.

Junto al diseño del modelo anterior, se requiere la implementación de una herramienta software en línea que interprete el modelo asociado a cualquier encuesta, que lo presente gráficamente al usuario para recoger sus respuestas, y que proporcione las respuestas recogidas al administrador.

De este modo, la parte central del problema abordado en este proyecto es la necesidad de implementar encuestas de satisfacción y a partir de ella se desarrolla la idea de implementar una aplicación que soporte múltiples encuestas, aportando dinamismo a la presentación de las preguntas en lugar de presentaciones globales de preguntas estáticas.

El INE hasta este momento no disponía de una aplicación de características similares, por lo que para cada encuesta que se proponía había que seguir un mecanismo individual y específico. La construcción de la herramienta propuesta supone no tener que construir un formulario específico por encuesta, reduciendo la carga de trabajo a un único paso que es la traducción de un listado de preguntas y opciones al esquema de base de datos de la herramienta.

## **1.2. Objetivos del proyecto**

El objetivo del proyecto es construir una aplicación informática que sirva como contenedor y administrador de encuestas así como de las respuestas introducidas por los usuarios para las futuras encuestas que desee realizar el INE en su página Web.

Esta aplicación tiene que ser capaz de administrar múltiples encuestas, para lo que cuenta con una estructura de base de datos diseñada para dar cabida a multitud de preguntas de diferentes tipos y encuestas, y para ofrecer la oportunidad de reutilizar preguntas y opciones en aquellos casos en las que éstas se repitan, dentro de la misma encuesta o entre varias encuestas.

Además, la aplicación deber permitir al usuario realizar encuestas relacionadas con su localización actual dentro del sitio Web oficial del INE. Esto se realizará mediante patrones que se aplican sobre cada encuesta con el fin de delimitar su área de funcionamiento, y conocer así la opinión sobre un servicio del que se tiene conocimiento.

Una novedad en el planteamiento seguido es que la aplicación también deberá ser capaz de controlar un flujo adaptativo para evitar mostrar preguntas innecesarias atendiendo a ciertas respuestas recibidas.

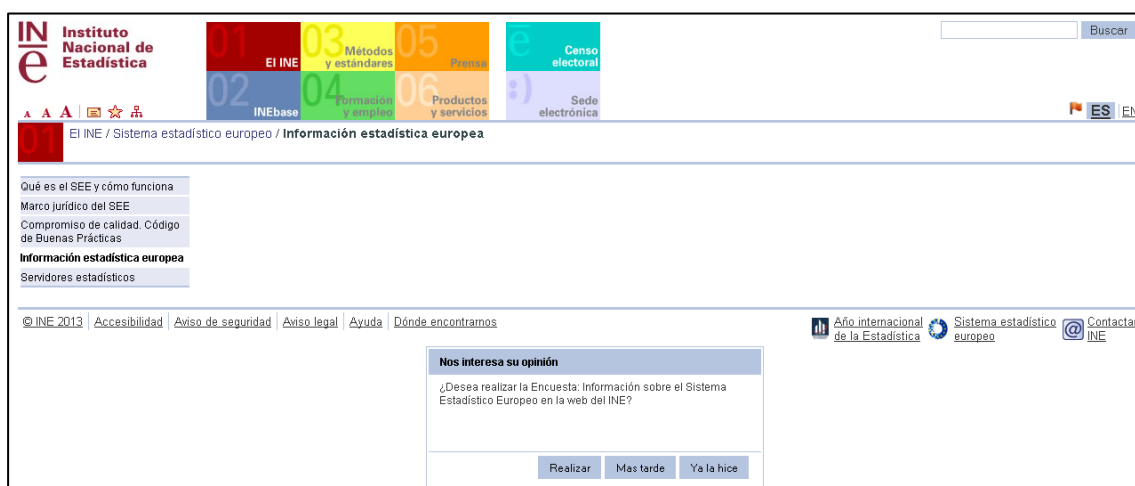
Por último, la aplicación deberá recoger todas las respuestas del usuario para almacenarlas en la base de datos con las que el INE podrá analizar resultados y generar estadísticas, siempre apoyándose en el esquema de respuestas diseñado.

### 1.3. Descripción general de la herramienta desarrollada

El proceso de uso de la herramienta desarrollada se divide en cuatro fases, las cuales se detallan a lo largo de todo el documento.

La primera fase es analizar la **localización del usuario** dentro del sitio Web del INE, y a partir de esa localización obtener una encuesta concreta. Mediante cookies<sup>1</sup> se obtiene si un usuario ha realizado ya un cuestionario. Antes de evaluar la cookie para saber si la encuesta ya ha sido realizada por el usuario, se comprueba que el usuario tenga las cookies habilitadas, y de lo contrario se le muestra un mensaje de aviso. La encuesta se obtiene mediante una petición con la URL<sup>2</sup> de la página en la que está el usuario al servicio Web de la aplicación, que devuelve el identificador de la encuesta si la hubiera.

La segunda fase consiste en mostrar al usuario la **disponibilidad de una encuesta** a realizar mediante un cuadro de diálogo, como el mostrado en la Figura 1, que pregunte al usuario si quiere realizar la encuesta en el momento, más tarde o bien no quiere hacerla.



*Figura 1. Cuadro de diálogo de disponibilidad de cuestionario*

La tercera fase contiene la mayoría del peso de la aplicación. En esta fase se realiza la **presentación de preguntas** en pantalla, se controla el flujo adaptativo, el flujo en sentido hacia atrás y la sustentación de respuestas y otros parámetros en sesión.

En las siguientes figuras se puede observar el número máximo de preguntas que quedan, similar a barra de progreso, botones para navegar en ambos sentidos (Figuras 3 y 4) y ejemplos de diferentes tipos de preguntas que se han implementado para la herramienta como por ejemplo,

<sup>1</sup> En el apartado 3.2.3, Gestión de la sesión en cliente: cookies, se explica el concepto de cookie y en el apartado 4.2.2, Gestión de cuestionarios, se explica la estructura de las cookies utilizadas por la herramienta.

<sup>2</sup> Uniform Resource Locator (URL), puntero a un recurso de la Web

de única respuesta en horizontal (Figura 2), subpreguntas con única respuesta horizontal (Figura 3) o pregunta con múltiple respuesta en vertical y opción abierta (Figura 4).

The screenshot shows the top navigation bar of the Instituto Nacional de Estadística (INE) website. Below the navigation bar, there is a survey title: "Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE". A progress indicator shows "Le quedan por responder como máximo 8 preguntas." The main question is: "1. ¿Considera útil la información de este apartado?". Below the question, it says "Por favor, puntúe de 5 (muy útil) a 1 (nada útil)." and provides five radio button options: 1, 2, 3, 4, and 5. A "Siguiente" button is located at the bottom right of the question area. The footer contains links for "INE 2013", "Accesibilidad", "Aviso de seguridad", "Aviso legal", "Ayuda", "Dónde encontramos", "Año internacional de la Estadística", "Sistema estadístico europeo", and "Contactar INE".

*Figura 2. Presentación de pregunta con opciones de tipo radio en horizontal*

The screenshot shows the same survey interface as Figure 2, but with a different question: "2. Indique el nivel de relevancia (interesante para sus necesidades) de la información incluida en este apartado." Below the question, it says "Por favor, puntúe de 5 (muy relevante) a 1 (nada relevante):". There are three sub-questions, each with five radio button options: "Información básica acerca del Sistema Estadístico Europeo", "Información sobre la calidad de las estadísticas europeas", and "Tablas estadísticas europeas más consultadas". Each sub-question also has a "No sabe / No contesta" option. A "Siguiente" button is located at the bottom right of the question area. The footer contains links for "INE 2013", "Accesibilidad", "Aviso de seguridad", "Aviso legal", "Ayuda", "Dónde encontramos", "Año internacional de la Estadística", "Sistema estadístico europeo", and "Contactar INE".

*Figura 3. Presentación de pregunta con subpreguntas y opciones tipo radio en horizontal*

The screenshot shows the same survey interface as Figure 2, but with a different question: "5. Principales razones por las que el nivel de confianza es alto." Below the question, it says "Por favor, seleccione un máximo de 2 respuestas de las siguientes:". There are five checkbox options: "Confío en los datos (por mi experiencia profesional)", "He oído/leído algo positivo sobre los datos", "Los datos son fáciles de entender", "Libre de interferencia política", and "Otras respuestas: (indique cuáles)". A text input field is provided for the "Otras respuestas" option. A "Siguiente" button is located at the bottom right of the question area. The footer contains links for "INE 2013", "Accesibilidad", "Aviso de seguridad", "Aviso legal", "Ayuda", "Dónde encontramos", "Año internacional de la Estadística", "Sistema estadístico europeo", and "Contactar INE".

*Figura 4. Presentación de pregunta con múltiples opciones*

Por último se insertan las **respuestas** recogidas en la base de datos de manera que una futura herramienta para analizar respuestas pueda extraerlas de forma sencilla. En este cometido el

diseño de la tabla de respuestas será altamente reutilizable. En el Anexo I se muestran el flujo completo de una encuesta de ejemplo administrada por la herramienta desarrollada.

## **2. Análisis de herramientas de edición de cuestionarios Web**

### **2.1. Descripción de herramientas existentes**

Durante del desarrollo del proyecto se ha realizado un análisis de sistemas similares a la herramienta desarrollada, ya existentes en la red. Para realizar la comparativa con la herramienta que se presenta se han seleccionado las siguientes tres herramientas:

#### **Google Forms**

Google Forms es una aplicación desarrollada por Google, dentro de la herramienta Drive, para crear tus propias encuestas de manera fácil y rápida, pero con un número de aportaciones inferior a las demás. Su coste gratuito es un punto a favor para utilizar esta herramienta. Google ofrece un portal para la creación de encuestas en el que el cliente crea páginas y a dichas páginas le añade preguntas de un tipo seleccionado y con un número de opciones asignadas, así como una administración de las respuestas en un fichero de Excel, generado al finalizar la creación del formulario e incluido en la propia cuenta de Google Drive, o bien en la pantalla de respuestas que ofrece la propia herramienta Google Forms.

Como se puede apreciar en la Figura 5, Google Forms utiliza su primera página Web como página de presentación, ya que su título se mantiene en todas las páginas.

The image shows a screenshot of a Google Form titled "Página 1". The form is displayed in two pages. The first page, "Página 1", contains two multiple-choice questions. The first question is "Primera pregunta de página 1" with options "Option 1" and "Option 2". The second question is "Segunda pregunta de página 1" with options "Option 1", "Option 2", and "Other:" followed by a text input field. A "Continue »" button is at the bottom. The second page, "Página 2", contains a single text input question titled "Primera pregunta de página 2". It has "« Back" and "Submit" buttons. A warning message "Never submit passwords through Google Forms." is visible. Both pages show the "Powered by Google Drive" logo and a disclaimer: "This content is neither created nor endorsed by Google. Report Abuse - Terms of Service - Additional Terms".

*Figura 5. Encuesta implementada en Google Forms*

## Eval & Go

Eval & Go es una herramienta para la realización y el soporte de encuestas. Sus aportaciones son similares a las ofrecidas por la herramienta de este proyecto. En su contra tiene que para gozar de sus servicios hay que pagar un alto precio, ofreciendo en modo gratuito características muy parecidas a las de Google Forms. Para llegar a construir una encuesta, Eval & Go sirve al cliente de un portal donde, de manera esquemática, crea las páginas incluyendo las preguntas que desee, creadas a partir del tipo de respuesta, y las opciones de cada pregunta. El resultado de crear una encuesta el portal mencionado se refleja en la Figura 6, donde se pueden ver aportaciones como diferentes tipos de pregunta, navegación en ambos sentidos (botones precedente y siguiente) y la barra de progreso.

The figure displays three sequential screenshots of the Eval & Go survey interface, illustrating different question types and navigation features. Each screenshot shows a header with the 'EVAL & GO' logo and the title 'Mi encuesta'. A progress bar at the top of each page indicates the current position in the survey.

- Page 1 / 3 (33 %):** The first question is 'Primera pregunta de prueba'. It features two radio button options: 'Opción 1' and 'Opción 2'. A 'Siguiente' (Next) button is located at the bottom right.
- Page 2 / 3 (67 %):** The second question is 'Segunda pregunta de prueba'. It features three radio button options: 'Opción 1', 'Opción 2', and 'Opción 3'. Navigation buttons 'Precedente' (Previous) and 'Siguiente' (Next) are located at the bottom left and right, respectively.
- Page 3 / 3 (100 %):** The third question is 'Tercera pregunta de prueba'. It features four checkbox options: 'Opción 1', 'Opción 2', 'Opción 3', and 'Opción 4', followed by an 'Otro' (Other) option with a text input field. Navigation buttons 'Precedente' (Previous) and 'Fin' (End) are located at the bottom left and right, respectively.

*Figura 6. Encuesta implementada en Eval & Go*

## E-encuesta

E-encuesta.com es otra herramienta que ofrece diseño y mantenimiento de encuestas y que, en relación con la herramienta que se presenta, también está implementada con struts. De la misma manera que Eval & Go, esta herramienta también ofrece servicios de pago, pero tiene un servicio gratuito con más funcionalidades que el anterior, y además cuenta con versiones de pago que ofrecen más funcionalidad a menor coste. En la Figura 7 se puede ver un ejemplo de encuesta generada por e-encuesta.com a través de su portal de creación de encuestas, con diferentes tipos de preguntas seleccionadas como son única opción en horizontal y en vertical, opción libre para el usuario y múltiple opción en vertical, y como e-encuesta.com no aporta barra de progreso.

The image displays two screenshots of a survey titled "Encuesta TFG" created on the e-encuesta.com platform. The top screenshot shows "1. Página 1" with the description "Sin descripción". It contains two questions: "1. Primera pregunta de página 1" with two radio button options ("Opción 1" and "Opción 2"), and "2. Segunda pregunta de página 1" with three radio button options ("Opción 1", "Opción 2", "Opción 3") and a text input field for "Otro (por favor, especifique)". Navigation buttons include "Precedente" and "Siguiendo", and a "Siguiendo >>" link. The bottom screenshot shows "2. Página 2" with question "3. Primera pregunta de página 2" featuring three checkbox options ("Opción 1", "Opción 2", "Otro (por favor, especifique)") and a text input field. Navigation buttons include "<< Anterior" and "Finalizar >>". Both pages include the "Powered by: e-encuesta.com" logo.

*Figura 7. Encuesta implementada en e-encuesta.com*

## 2.2. Limitaciones de las herramientas similares

**Google Forms** es una alternativa gratuita para construir formularios y recoger respuestas, pero tiene algunas limitaciones que se comentan a continuación. Esta herramienta toma una interpretación diferente de pregunta: construye páginas en lugar de preguntas, y son esas páginas las que tienen preguntas, concepto cercano al de la herramienta presentada en este proyecto sobre preguntas y subpreguntas.

Tiene como limitaciones el no poder dar un estilo propio a la aplicación, el tener capacidad máxima de su hoja de Excel habilitada para las respuestas de un máximo de 250 columnas (1 por pregunta) o 400.000 celdas, el no presentar una barra de progreso, el no tener servicio multilingüe y el no soportar flujos normal y adaptativo a la vez. Está más orientada a simples formularios o pequeños cuestionarios personales que a la realización de encuestas de satisfacción más complejas.

**E-encuesta** tiene una gran similitud con la herramienta construida ya que está implementada usando el mismo framework, Struts. En su versión gratuita tiene varias limitaciones como no aceptar el flujo adaptativo ni la navegación en ambos sentidos, no permitir el cierre programado de una encuesta, tener estilo perteneciente a un grupo de preestablecidos y no ofrece servicio multilingüe.

**Eval & Go** es la opción que, en su versión gratuita, presta menos servicios. No ofrece servicio multilingüe, no tiene flujo adaptativo ni navegación en ambos sentidos, no ofrece página de agradecimiento ni página de introducción, tiene un número de respuestas/mes limitado y aún con la versión de pago no permite el flujo adaptativo hacia atrás. Como aporte positivo, ofrece una gran variedad de preguntas y muestra la barra de progreso de la encuesta como se ve en la Figura 6.

## 2.3. Aportaciones de la herramienta desarrollada

La aplicación desarrollada ofrece al administrador de las encuestas, el INE en este caso, varias de las aportaciones de las 3 herramientas que se han descrito en la sección anterior, así como de otras nuevas:

- Posibilidad de mostrar encuestas multilingües sin tener que crear una por cada idioma. El mecanismo utilizado para crear encuestas multilingües se expone en el Anexo II.
- Mostrar una página de introducción y otra de agradecimiento al usuario.
- Recorrer la encuesta en cualquier momento y en cualquier sentido manteniendo siempre las respuestas ya marcadas a menos que el usuario decida cambiar alguna de ellas.
- Disponer de un flujo alternativo de preguntas para evitar al usuario preguntas innecesarias llevándole a las realmente importantes, adaptando en todo momento las futuras preguntas a sus actuales respuestas.
- Dotar de una página Web de instrucciones previa a las preguntas.
- Programar una encuesta con fecha de inicio y/o fin, y reutilizar preguntas y opciones dentro de la misma encuesta o entre diferentes encuestas con el objetivo de ahorrar el máximo espacio posible en base de datos.

### **3. Tecnologías empleadas**

En este apartado se introducen las tecnologías que se han utilizado en el desarrollo de la herramienta, tanto en la parte servidor como en la parte del cliente o navegador.

#### **3.1. Tecnologías de la parte servidor**

Las tecnologías utilizadas en la parte servidor se introducen a continuación, y entre ellas figuran JavaServlets<sup>3</sup>, JSP<sup>4</sup>, Apache Struts II<sup>5</sup>, SSI<sup>6</sup> y la comunicación entre cliente y servidor utilizando HTTP Request y HTTP Response.

##### **3.1.1. Java Servlets**

La tecnología **Java Servlets** provee al desarrollador Web de una herramienta simple y consistente para extender la funcionalidad de un servidor Web con Java, y representa paradigma de petición/respuesta en la relación con el lado. Un servlet es un programa que recoge una petición del cliente y realiza alguna lógica de negocio en el lado del servidor para finalmente devolver un resultado en forma de página al cliente, y pueden ser cargados cuando el servidor Web lo necesite. JSP es la tecnología que se utiliza en este proyecto para representar gráficamente los resultados de los servlets.

Esta tecnología se ejecuta sobre una máquina virtual de Java (Java Virtual Machine, JVM) lo que la convierte en una solución muy portable, y necesita de un contenedor de servlets, también llamado motor de servlet. El contenedor de servlets trabaja acoplado a una instancia del servidor Web para tratar las solicitudes de servicio. Para la realización de este proyecto se ha utilizado como servidor Web Apache Tomcat<sup>7</sup>, que también está implementado en Java y corre sobre una máquina virtual de Java de igual manera y cuyo complemento empleado como contenedor de servlets recibe el nombre de Catalina.

La función de los servlets es realizar el proceso asignado a una capa intermedia dejando al servidor encargado de dirigir las conexiones de red, la negociación del protocolo y el tipo de carga entre otras cosas. En un modelo de tres capas los servlets actúan como un servidor de aplicación situándose entre el navegador y el tratado de los datos, y liberan de carga tanto al cliente como al servidor encargándose de determinados procesos.

A continuación se presenta como se crea un servlet en el contenedor Catalina mencionado anteriormente y el ciclo de vida que sigue dicho servlet durante la ejecución de la herramienta.

##### **Instanciación y carga en memoria**

La instanciación y carga en memoria se produce tras levantar un servidor de aplicaciones que soporte servlets y JSP, como Apache Tomcat. Este servidor de aplicaciones lee el descriptor de despliegue de la aplicación a la que pertenece, lo que implica la creación, por parte del

---

<sup>3</sup> JavaServlets, <http://www.oracle.com/technetwork/java/index-jsp-135475.html>

<sup>4</sup> Java Server Pages, <http://www.oracle.com/technetwork/java/javaee/jsp/index.html>

<sup>5</sup> Apache Struts II, <http://struts.apache.org/development/2.x/>

<sup>6</sup> Server Side Includes, [http://msdn.microsoft.com/en-us/library/ms525185\(v=vs.90\).aspx](http://msdn.microsoft.com/en-us/library/ms525185(v=vs.90).aspx)

<sup>7</sup> Apache Tomcat, <http://tomcat.apache.org/>



contenedor, de un objeto servlet por cada elemento `<servlet>` que contenga el descriptor. El descriptor de despliegue de una aplicación, conocido como `web.xml` en todas las aplicaciones Web dinámicas y cuyo esquema simple se puede ver en la Figura 8, es un fichero que incluye los parámetros de configuración que indican al servidor de aplicaciones que servlets se van a utilizar indicados por la etiqueta `<servlet>`, los nombres indicados por la etiqueta `<servlet-name>`, sus clases asociadas representadas por `<servlet-class>` o el mapeo de URLs que va a llevar a cada uno de ellos entre las etiquetas `<servlet-mapping>` y otros parámetros de configuración que incluye este tipo de fichero.

```
<web-app xmlns="http://java.sun.com/xml/ns/javaee" version="2.5">
  <servlet>
    <servlet-name> MiServlet </servlet-name>
    <servlet-class> MiSitio.MiServidor.MiServlet </servlet-class>
  </servlet>
  <servlet-mapping>
    <servlet-name> MiServlet </servlet-name>
    <url-pattern> /* </url-pattern>
  </servlet-mapping>
</web-app>
```

*Figura 8. Esquema fichero web.xml*

El servidor de aplicaciones creará cada servlet mediante el método estático de `java.lang.Class` `forName(String s)` y el método `newInstance()` de `Class`. Como argumento del método `forName(String nombre)` se utiliza el nombre del fichero `‘.class’` obtenido tras compilar el servlet. Esta información se obtiene del valor especificado en la etiqueta xml `<servlet>` de un descriptor de despliegue como el representado en la Figura 8.

### **Ciclo de vida**

Cuando el servlet ya ha sido instanciado, pasa a lo largo de su tiempo instanciado por 3 etapas que son la inicialización, los métodos de servicio y la destrucción o finalización, etapas que se detallan a continuación.

- **Inicialización**: Esta fase implica la ejecución única del método `init(ServletConfig config)`. Un servlet se puede inicializar en tres momentos diferentes dependiendo de la configuración del servidor, ya sea al arrancar el servidor, cuando el servlet reciba la primera petición HTTP antes de ejecutar el método de servicio o bien a petición del administrador del servidor.
- **Métodos de servicio**: Un servlet pasa por esta fase cada vez que recibe una petición HTTP, lo que provoca la ejecución de un método del tipo *service* (`ServletRequest request, ServletResponse response`), siendo *service* el nombre del método dependiendo del atributo `‘method’` del formulario o el método de la cabecera de la petición. En esta aplicación se utilizan los tipos de envío GET y POST luego el nombre de los métodos de la clase `Servlet` que se ejecutan son `doGet (ServletRequest request, ServletResponse response)` y `doPost(ServletRequest request, ServletResponse response)`.

- Finalización: En esta última etapa se destruye la instanciación del servlet. Se ejecuta una única vez el método `destroy()`, cuyo encargado de invocarlo es el contenedor de servlets del servidor, que deja fuera de servicio al servlet hasta que vuelva a ser cargado/instanciado e inicializado.

### 3.1.2. Java Server Pages (JSP) y JSP Standard Tag Library (JSTL)

**Java Server Pages (JSP)** es una tecnología usada para construir aplicaciones Web que presenta contenido de manera dinámica. JSP forma parte del contenedor de elementos del lado del servidor.

Una página JSP se forma a partir de contenido estático HTML o componentes XML, etiquetas propias de JSP (por ejemplo JSTL) y de manera opcional scriptlets que son líneas de código Java puro integradas en el fichero JSP. Esta tecnología forma la capa de presentación en cualquier modelo de arquitectura Web de N capas.

JSP tiene permitido definir objetos dentro de un ámbito definido a elegir entre la página, la petición (request), la sesión o bien la aplicación. También permite el acceso a JavaBeans y sus atributos a los que previamente se les deben haber declarado métodos de acceso público a sus atributos.

Las páginas JSP son compiladas en servlets por lo que se dispone de mayor potencia para realizar las operaciones. Además estas páginas son portables a la mayoría de servidores Web. En comparación con otros lenguajes de generación dinámica de contenido como Javascript, este se limita a generar código en el lado del cliente sin poder acceder a recursos del servidor a los que sí pueden acceder las páginas JSP, y en comparación con SSI (Server Side Includes), este se dedica para declaraciones sencillas y no para operaciones de peso.

Las páginas JSP de la herramienta utilizan etiquetas de la librería **JSP Standard Tag Library<sup>8</sup>** (JSTL), que es un conjunto de librerías con funciones comunes utilizadas desde páginas JSP. Estas funciones sirven para sustituir trozos de código java embebido, llamados scriptlets, que resultan más difíciles de programar y que por sí solos no contribuyen al código HTML.

La librería de JSTL utilizada en la realización de este proyecto ha sido la de etiquetas core, que contiene, entre otras, las siguientes funciones que han servido de ayuda en la implementación de las páginas JSP:

- `<c:choose>`: Esta etiqueta representa un condicional múltiple. Compuesto por `<c:when>` y `<c:otherwise>`.
- `<c:when>`: Esta etiqueta representa un caso dentro de un condicional `<c:choose>`, cuya condición la establece el atributo de `<c:when>` llamado “test”.
- `<c:otherwise>`: Esta etiqueta representa el caso por defecto dentro de un condicional `<c:choose>`
- `<c:set>`: Esta etiqueta tiene como funcionalidad asignar valor a una expresión. Se le puede asignar el valor de una variable en sesión o un valor nuevo.

---

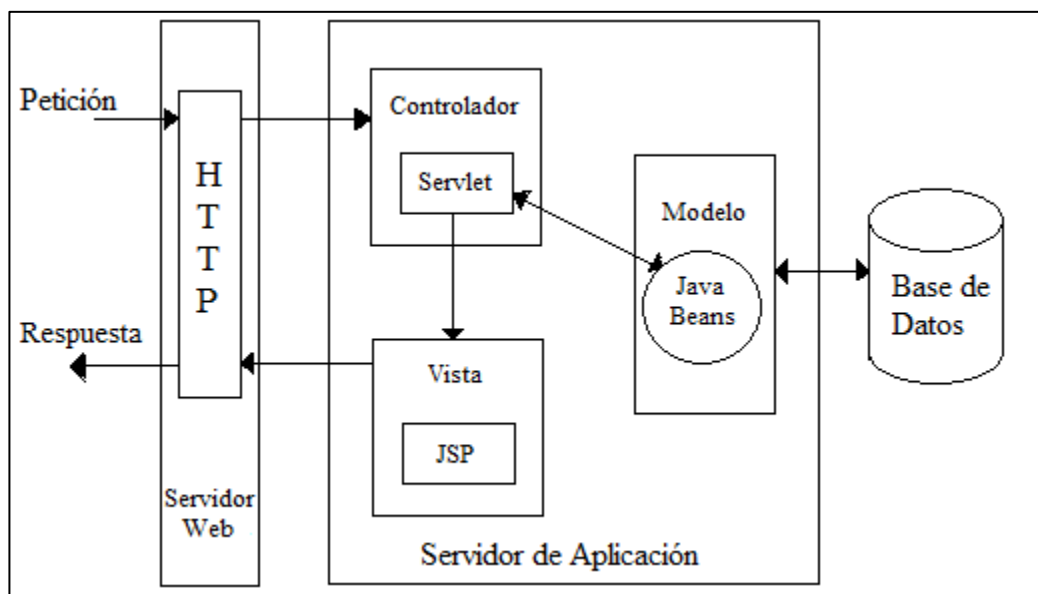
<sup>8</sup> JSTL, <http://www.oracle.com/technetwork/java/index-jsp-135995.html>

- `<c:forEach>`: Esta etiqueta representa un bucle. Los atributos internos de esta etiqueta, como “items”, “begin”, “var” o “end”, indican que contador o array de elementos hay que utilizar durante la ejecución del bucle.
- `<c:if>`: Esta etiqueta representa un condicional simple. Únicamente se ejecuta su contenido si se cumple la condición del atributo “test”.
- `<c:import>`: Esta etiqueta se utiliza para importar otras páginas jsp, indicando en sus atributos “URL”, “context”, “charEncoding”, “var”, “scope” y/o “varReader” las propiedades del documento a importar.

### 3.1.3. Apache Struts II

**Apache Struts II** es un marco basado en Java desarrollado por Apache cuyo fin es la construcción de aplicaciones Web.

Sin ser una tecnología en sí misma, Struts tiene como núcleo las tecnologías Java Servlets y JSP siguiendo el patrón arquitectónico MVC (Modelo-Vista-Controlador), reflejado en la Figura 9, que fue desarrollado inicialmente para trabajar en el espacio de interfaces gráficas de usuario (GUI)



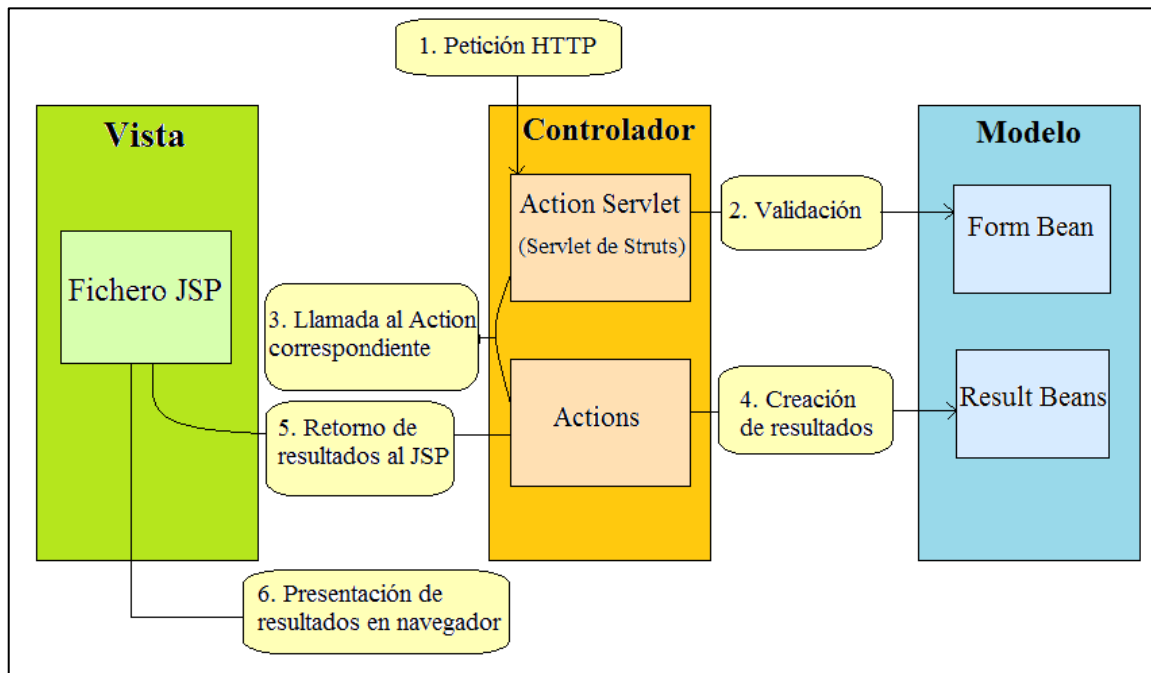
*Figura 9. Modelo MVC en Aplicación Web Genérico*

Los servlets son óptimos realizando la lógica de negocio, pero a la hora de presentar los resultados la tecnología JSP es mejor.

El modelo MVC permite la reutilización de componentes y el cambio en la implementación de una capa afecta mínimamente al resto de capas. Este es el principal beneficio de Struts.

El flujo de control en Struts II que especifica el modelo MVC para esta tecnología, se sostiene en tres componentes que son la vista, el controlador y el modelo, cada uno de ellos con sus propios elementos, y que interactúan entre ellos siguiendo los pasos ilustrados en la Figura 10. El primer componente de la ilustración en arrancar es el controlador, que recibe la petición HTTP para, a continuación, validar el formulario de la petición y a partir del resultado, invocar

al Action correspondiente para que genere resultados que se envían al fichero JSP para que los represente y envíe la respuesta al navegador. En la Figura 10 se muestran componente por componente los elementos de los tres principales bloques: Controlador, Modelo y Vista.



*Figura 10. Modelo MVC específico de Struts*

A continuación se detallan los componentes de una aplicación Web dinámica que utiliza el framework Struts y como se configuran cada uno de ellos.

### **Componentes del controlador de Struts**

#### **ActionServlet**

Este componente extiende la clase `javax.servlet.http.HttpServlet` y es el responsable de recibir y redirigir el tráfico de peticiones HTTP a los distintos manejadores del entorno.

El `ActionServlet` no es una clase abstracta y se configura en el descriptor de despliegue llamado `web.xml` incluido en la ruta `/WEB-INF`. En este fichero se indica el nombre del servlet, la clase a la que se refiere y el mapeo a realizar con los servlets. Para el servlet proporcionado por Struts de nombre `org.apache.struts.action.ActionServlet` se le indica en el fichero de configuración que para toda petición del cliente recibida con patrón `*.do` se dirija al fichero de configuración específico de Struts (`struts-config.xml`) donde se resuelve la clase ayudante que realiza la lógica de negocio en conjunto con el modelo interaccionando con él. Esa clase ayudante extiende de la clase `org.apache.struts.action.Action`.

#### **Action**

Este componente extiende de la clase `Action` original del paquete desarrollado por el proyecto Apache. Es el propio Struts el que maneja los `Action` creándolos y reutilizándolos por lo que no es una clase que deba instanciar atributos de la propia clase ya que sus datos no se mantienen. Un componente de tipo `Action` debe sobrescribir el método `execute` declarado en la superclase original.

```
public ActionForward execute(ActionMapping mapping,  
    ActionForm form,  
    HttpServletRequest request,  
    HttpServletResponse response);
```

En este método, en el caso de que el flujo venga previa de los datos se recogen los datos de la clase extendida de ActionForm particular gracias al parámetro de dicho tipo genérico. También se lleva a cabo el proceso de la acción con esos datos y se prepara la respuesta a dicha acción.

El método *execute* devuelve un elemento de tipo ActionForward que indica la página a mostrar como respuesta. Este forward se localiza gracias al método *find(String forward\_name)* del elemento de la clase ActionMapping que recibimos como parámetro. Dicho *forward\_name* se corresponde con el nombre del forward declarado en ese action en el fichero de configuración struts-config.xml de la siguiente forma.

La configuración de una aplicación Web dinámica que implementa Struts se almacena en el fichero struts-config.xml y que acompaña al fichero de configuración web.xml común para las aplicaciones Web. En este fichero se incluyen las reglas para el mapeo de los actions, los actionforms, los forwards comunes o específicos de cada action o la definición de excepciones. El formato utilizado para definir todos estos objetos y reglas es XML.

Las reglas para encontrar que action es el encargado de manejar la entrada de una petición, <action-mappings>, se declaran de la siguiente manera:

```
<action-mappings>  
    <action  
        path="/ruta2"  
        input="/ruta1"  
        scope="request|session"  
        name="NombreActionForm"  
        validate="true|false"  
        type="ruta_paquete.NombreAction">  
        <forward name="Nombre_forward" path="paginas/pagina.jsp">  
        <forward name="Nombre_forward2" path="paginas/pagina2.jsp">  
    </action>  
</action-mappings>
```

A continuación se explica el significado de cada una de las propiedades de la configuración de un action:

- path: indica la ruta URI de la petición necesaria para seleccionar dicho mapeo.
- input: indica la ruta URI de la página/acción que debe mapear si la validación del formulario devuelve algún error.
- type: especifica la ruta de la clase Java con la implementación del action mapeado.
- scope: establece el ámbito de los datos del formulario. Sus valores pueden ser 'request' o 'session'.
- validate: especifica si debe validarse el formulario mediante un ActionForm antes de ejecutarse el action. Sus valores pueden ser 'true' o 'false'.
- name: indica el nombre del componente ActionForm asociado a este action.
- forward:
  - name: nombre del forward. Se utiliza su valor para hacer referencia a él desde el action.
  - path: ruta de página o action a mapear de destino.

### **Componentes del modelo de Struts**

Como componentes del modelo de Struts podemos incluir todos los JavaBeans necesarios para dar estructura a la aplicación. En este bloque no entran los elementos de tipo Action, que aunque si realizan parte de lógica de negocio e interaccionan con el modelado, realizan también acciones de control por lo que se les considera componentes de controlador.

También pertenecen al modelo los componentes de tipo ActionForm, encargados de validar los resultados del formulario enviado.

### **ActionForm**

Las etiquetas para definir los componentes que actúan como ActionForm son las siguientes:

```
<form-beans>
  <form-bean
    name=" NombreActionForm"
    type="ruta_paquete.NombreActionForm">
    <form-property
      name="propiedad"
      type="java.lang.Long"/>
    </form-bean>
  </form-beans>
```

- name: el nombre del ActionForm con el que nos vamos a referir a él en *<action-mappings>*.
- type: la ruta del paquete y el nombre de la clase que extiende de ActionForm y que realiza las funciones de comprobación del formulario.
- form-property:
  - name: contiene el nombre de una propiedad del form.
  - type: especifica el tipo de dato de la propiedad.

### **Componentes de la vista de Struts**

Los componentes del bloque de vista de este framework son los Java Server Pages (JSP) cuyo funcionamiento se ha detallado previamente en el punto 3.1.2 (Java Server Pages (JSP) y JSP Standard Tag Library (JSTL)).

#### **3.1.4. Server Side Includes (SSI)**

**Server Side Includes (SSI)** son directivas que se sitúan en páginas HTML y que son evaluadas en el servidor cuando las páginas se cargan. Permiten añadir contenido generado dinámicamente a la página HTML existente sin tener que generar dinámicamente la página al completo.

Esta tecnología se utiliza en este proyecto para incluir dinámicamente los ficheros necesarios en la cabecera de la página Web del INE.

#### **3.1.5. Comunicación cliente-servidor: HTTP Request y HTTP Response**

La comunicación HTTP entre cliente y servidor se basa en un modelo de solicitud/respuesta. A lo largo de esta aplicación hay numerosas comunicaciones entre el navegador y el servidor siendo la primera de ellas al solicitar al servlet las encuestas disponibles para un patrón, y el resto durante la encuesta, estableciéndose una por cada pregunta.

La solicitud **HTTP Request** es un mensaje cuya primera línea de salida sigue siempre el siguiente formato:

método / URI / v\_http

Donde *método* es el método HTTP utilizado, *URI* es el identificador del recurso que se quiere obtener y *v\_http* es la versión del protocolo utilizada.

Los métodos HTTP indican la acción que se desea realizar sobre el recurso indicado. El protocolo HTTP define los siguientes 8 métodos: HEAD, POST, GET, PUT, DELETE, TRACE, OPTIONS y CONNECT. En esta aplicación se utiliza el método GET para conectar con el servlet que nos devuelve las encuestas disponibles, y el método POST para enviar las respuestas al servidor. A continuación de la línea de petición se pueden incluir cero o más campos en la cabecera, tales como *Accept* (tipos que acepta), *Accept-Charset* (conjunto de caracteres aceptables para la respuesta), *Accept-Encoding*, *Accept-Language*, *Content-Encoding* (codificación del contenido), *Content-type* (tipo del cuerpo del mensaje enviado al destinatario),

Host (nombre del host y puertos que se utilizan para obtener los recursos), *Referer* (especificar la ruta de los recursos desde la que se obtiene la respuesta) y/o *User-Agent* (información del cliente que origina la petición).

Tras recibir una solicitud, el servidor prepara un mensaje de respuesta **HTTP response** para el cliente. Esta respuesta esta formada por una primera línea de estado con la versión HTTP, un código numérico y el texto explicativo de dicho código, seguida de campos de cabecera similares a los de la cabecera de solicitud y el código del recurso requerido. Un ejemplo de línea de estado en una respuesta HTTP es:

```
HTTP/1.1 200 OK
```

## 3.2. Tecnologías de la parte cliente

En este apartado se mencionan las tecnologías que utiliza la herramienta en la parte cliente, y se da una breve introducción de los conceptos básicos de éstas. Entre las tecnologías utilizadas en la parte cliente figuran JavaScript<sup>9</sup>, JSON<sup>10</sup> y cookies.

### 3.2.1. JavaScript

**JavaScript** es un lenguaje de programación interpretado incrustado en el navegador de manera que realiza sus acciones sin necesidad de comunicar con el servidor de la aplicación Web y permite interactuar con el lenguaje HTML generado. Una de sus principales ventajas es que es rápido y que al ser interpretado no se necesita recompilar los archivos ante cualquier acción realizada por una función de JavaScript ni los propios ficheros ‘.js’ (extensión de fichero JavaScript) cuando estos se modifican.

Algunas de sus utilidades principales que se aprovechan en la realización de este proyecto son las de realizar comprobaciones de datos insertados en formularios, cambios en el formulario de manera dinámica y/o efectos gráficos.

Para la implementación de efectos gráficos nos ayudamos de la librería de JavaScript denominada **JQuery UI**<sup>11</sup> incluida en la librería principal JQuery<sup>12</sup> cuya implementación es de código abierto. Nos serviremos de las ventajas que ofrece JQuery a la hora de presentar al usuario la opción de realizar la encuesta en un cuadro de diálogo. Además JQuery UI aporta un fichero CSS para dar estilo a los componentes utilizados, que va a ser modificado para seguir el estilo común del sitio Web del INE.

---

<sup>9</sup> JavaScript, <http://www.w3.org/standards/Webdesign/script>

<sup>10</sup> JavaScript Object Notation, <http://www.json.org/>

<sup>11</sup> JQuery.UI, librería perteneciente a la biblioteca JQuery, aporta funcionalidad específica en la creación de efectos visuales, interacciones y widgets.

<sup>12</sup> JQuery es una biblioteca de javascript cuyo código es libre y abierto y que aporta funcionalidad para la creación de interfaces, manejo de eventos, animaciones y el uso de Ajax.



### 3.2.2. JavaScript Object Notation (JSON)

**JavaScript Object Notation (JSON)** es un formato ligero de intercambio de datos entre el cliente y el servidor. Es independiente de un lenguaje específico (la mayoría de ellos lo soportan), no tiene espacios de nombres, no tiene validador y no es extensible.

Su estructura es fácil de interpretar y generar por una máquina. Esta está constituida por dos subestructuras:

- Colección de pares nombre/valor.
- Una lista ordenada de valores.

Su presentación se implementa de la siguiente manera:

- Caso de par nombre / valor:  
`{ nombre1 : valor1 , nombre2 : valor2 , ... , nombreN : valorN }`
- Caso de un array de valores:  
`[ valor1 , valor2 , ... , valorN ]`

Comparándolo con otros lenguajes de intercambio de datos como XML, su principal rival, JSON tiene una gramática más pequeña y mapea más rápido las estructuras. Además, ocupa menos espacio que XML<sup>13</sup>, no necesita de ‘parseadores’ personalizados, es más compacto y permite incluir arrays.

En este proyecto nos servimos de JSON para la selección de la encuesta adecuada a partir de la URL en la que permanece el usuario. Con JavaScript realizaremos la llamada al servicio para que nos devuelva el resultado en el formato JSON. En el lado del servidor no se analiza ningún fragmento JSON, únicamente se genera la cadena resultante para la petición enviada. JavaScript cuenta con funciones incluidas en las librerías estándar como la función *getJSON()* la cual utilizamos para parsear el fragmento recogido como resultado.

### 3.2.3. Gestión de la sesión en cliente: cookies

Las **cookies** son estructuras de datos que contienen información recogida por un sitio Web la cual se encarga de almacenar en el navegador del usuario. Tiene como utilidad poder almacenar información para utilizarse posteriormente como si se tratase del ámbito de la sesión pero con los inconvenientes de que la información que contiene está al alcance del usuario pudiendo borrarlas o desactivarlas en cualquier momento y que cualquier servidor al que el cliente esté conectado puede acceder al contenido de todas las cookies del navegador.

La estructura básica de una cookie es la siguiente:

- Nombre
- Valor: atributo que representa el contenido de la cookie

---

<sup>13</sup> Extensible Markup Language, <http://www.w3.org/XML/>

- Caducidad: atributo que representa la fecha en la que expira la cookie. Existen varios tipos de cookies clasificándolas según su fecha de caducidad:
  - Sesión
  - Permanentes
  - Fecha personalizada
  - Con una fecha anterior a la actual. Se utiliza para borrar la cookie
- Dominio: atributo que representa los dominios donde la cookie tiene validez.
- Ruta: atributo que representa los directorios de dichos dominios donde la cookie tiene validez.
- Seguro: campo de la cookie que no tiene valor y únicamente indica si la cookie se transmite por canal seguro.
- HttpOnly: este atributo tampoco tiene valor e indica que dicha cookie sólo se puede transmitir por un protocolo HTTP. Dado que en este proyecto las cookies se tratan con JavaScript no se utiliza este atributo.

## **4. Arquitectura de la herramienta desarrollada**

En este apartado se introduce la arquitectura utilizada para desarrollar la herramienta, y tras ello se detallan los componentes que figuran en la arquitectura.

### **4.1. Diagrama de arquitectura**

En la Figura 11 se muestra la arquitectura de la herramienta desarrollada. Incluye dos controladores que representan los bloques de extracción de encuesta apropiada y control de la misma. Existe una base de datos a la que se conectan ambos controladores y un conjunto de modelos que se utilizan para representar los datos extraídos de la base de datos.

En la misma figura se da una breve descripción de la funcionalidad de cada uno y la comunicación entre ellos.

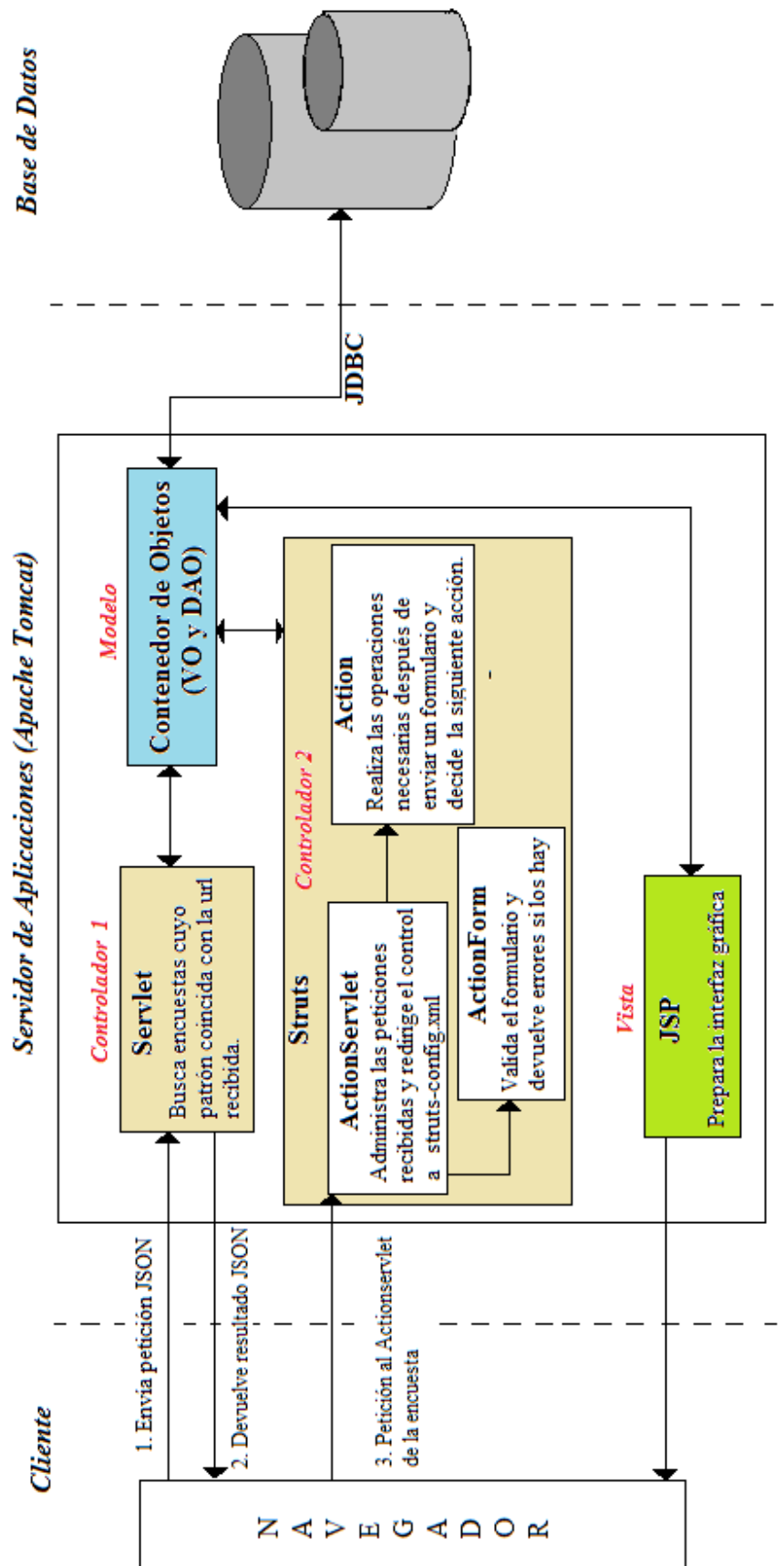


Figura 11. Diagrama de la arquitectura del sistema desarrollado

## 4.2. Descripción de los módulos de la arquitectura

En este apartado se detalla el modo de acceso a la base de datos y la comunicación que tiene con los controladores, y como se utilizan los controladores, el modelo y la interfaz gráfica para realizar la gestión de los cuestionarios.

### 4.2.1. Acceso a bases de datos

La lógica aplicada a la gestión de cuestionarios en el módulo de base de datos se resuelve en un esquema de datos relacional construido en base a las necesidades de la aplicación e implementada sobre una base de datos de **Oracle Database**, cuya administración y desarrollo se lleva a cabo mediante el cliente de bases de datos *Toad for Oracle* y el lenguaje **PL/SQL**<sup>14</sup>. El objetivo del esquema diseñado es que el peso de la lógica implementada en la parte de modelo sea lo más ligero posible aprovechando la potencia y capacidad de la base de datos de la que se dispone. El esquema de la base de datos se divide en distintos bloques que cumplen las distintas funciones que se han enumerado en la descripción general de la herramienta:

- Estructuración de las encuestas, preguntas y subpreguntas de cada encuesta, y opciones por preguntas y encuestas.
- Control del flujo de las preguntas por encuesta según la opción.
- Almacenamiento de las respuestas seleccionadas/insertadas por el usuario.

Además se incluye un bloque extra cuyos beneficios se aprovechan en el global de la aplicación. Este bloque aporta la característica de multilenguaje al aplicativo dado que el sitio Web del INE puede ser visitada en diferentes idiomas.

Como se muestra en la Figura 11, mediante la **API JDBC**<sup>15</sup> se realiza la conexión entre el servidor de aplicaciones y la base de datos, y se ejecutan operaciones, ya sean procedimientos almacenados, funciones o inserciones, a la base de datos en la que tenemos almacenado el esquema de la herramienta.

Para transformar el contenido de la base de datos a **JavaBeans** se utilizan clases java cuyo nombre va acompañado del identificador DAO (Data Access Object). Toda la negociación con la base de datos por parte de la herramienta se lleva a cabo en estos componentes. Además desde estos módulos se da forma a los objetos VO (Value Object) que representan cada objeto con sus atributos.

En la herramienta desarrollada el acceso a base de datos se realiza en dos ocasiones. La primera de ellas se lleva a cabo al llamar al Action principal con el path *index.do*. Este Action se utiliza al inicio de la encuesta y cada vez que se avanza de pregunta pero el path entonces cambia a *siguiente.do*. Por lo tanto, la primera vez que se llama a este Action al iniciar la encuesta, se realiza una conexión a base de datos en la que se extraen la encuesta a realizar y todas las preguntas con sus respectivas opciones, tipos y el flujo que las controla. A partir de los datos obtenidos se construye el modelo de datos de la encuesta.

---

<sup>14</sup> Procedural Language / SQL, <http://www.oracle.com/technetwork/database/features/plsql/index.html>

<sup>15</sup> Java Database Connectivity, <http://www.oracle.com/technetwork/java/overview-141217.html>

La segunda conexión se realiza al finalizar la encuesta para insertar las respuestas en base de datos traduciendo el modelo VO de respuestas al modelo de las mismas en el esquema.

El aspecto de la seguridad es muy importante en la encuesta ya que por medio del método de inyección SQL (**SQL Injection**) un usuario puede realizar operaciones no deseadas contra la base de datos como por ejemplo borrar o alterar datos del esquema. La inyección SQL consiste en insertar en cuadros de texto código SQL de carácter dañino para el propietario de la herramienta. Esa vulnerabilidad se puede solucionar formando las sentencias SQL a ejecutar en los DAOs de manera que sean seguras utilizando la clase **PreparedStatement** que proporciona Java y que forma las sentencias parametrizando los valores que hayan sido insertados en cuadros de texto. El driver JDBC automáticamente aplica el uso de secuencias de escape a los valores parametrizados.

#### 4.2.2. Gestión de cuestionarios

La gestión de cuestionar se lleva a cabo en dos bloques, que son la selección del cuestionario a realizar a partir de la URL y la generación y control del mismo por parte de la herramienta. La relación con la arquitectura de la herramienta en ambos bloques se detalla a continuación.

##### **Selección de cuestionarios**

Este módulo realiza las operaciones necesarias para obtener cual es la encuesta adecuada para mostrar al usuario según la página del sitio Web del INE en la que está en ese instante, tal y como se refleja en el recuadro de este módulo en la Figura 11.

Al cargar dicha página, se da forma en el lado del cliente y mediante Javascript a la URL de petición al servlet incluyendo como parámetro la URL de la página que se está cargando. El servlet en ese momento interpreta la llamada realizando una consulta a la base de datos de la que extrae una encuesta relacionada con el patrón que cumpla la cadena de caracteres que contiene el parámetro con la URL de la página y sus fechas de inicio y caducidad. En ningún caso se obtiene más de una encuesta por página dado que las URLs de las diferentes páginas del sitio Web contienen algún identificador único que ha sido utilizado para elaborar los patrones.

Antes de elaborar la respuesta en formato JSON, el servlet evalúa si la encuesta está vigente en ese momento, para lo cual la fecha actual tiene que estar en el período temporal entre la fecha de inicio y de caducidad. En el caso de que dicha condición se cumpla se devuelve una cadena con el siguiente formato:

```
{ Id: identificador , Nombre: título de la encuesta }
```

En el caso de que no exista encuesta o exista pero no esté vigente, se devuelve una cadena con el valor de '**Id**' igual a '**0**'.

Una vez haya devuelto el servlet, la cadena resultante en formato JSON se evalúa en el lado del cliente por medio de Javascript si el usuario cumple las condiciones para que se le muestre el cuadro de diálogo con la proposición de realizar la encuesta. Esta condición es que o bien no se tenga constancia de que el usuario hubiere realizado la encuesta, o bien hubiere rehusado de realizarla.

¿Cómo saber si el usuario ha realizado la encuesta o ha rehusado de hacerla? Mediante Javascript se recoge el valor de una cookie con el formato “**EncuestaINE**+‘*Identificador\_de\_encuesta*’”. Esta cookie, en caso de existir puede tener 2 valores:

- ‘1’: situación indicativa de que la encuesta ya ha sido realizada o denegada. En el caso de que el usuario seleccione que se le muestre más tarde también tiene este valor pero durante un tiempo limitado y tras ese período pasa a valer ‘0’.
- ‘0’: situación indicativa de que el usuario seleccionó hacerla más tarde y ese período ha finalizado.

En caso de que no exista la cookie o valga ‘0’ la encuesta no se ha realizado y por lo tanto se muestra el cuadro de diálogo para que el usuario decida si desea realizarla. La cookie creada por haber realizado la encuesta no tiene fecha de expiración y por lo tanto desaparece si el usuario las elimina manualmente. A partir de ese momento si el usuario vuelve a pasar por la misma página se le vuelve a presentar la posibilidad de hacer la encuesta al no haber constancia de que la haya realizado.

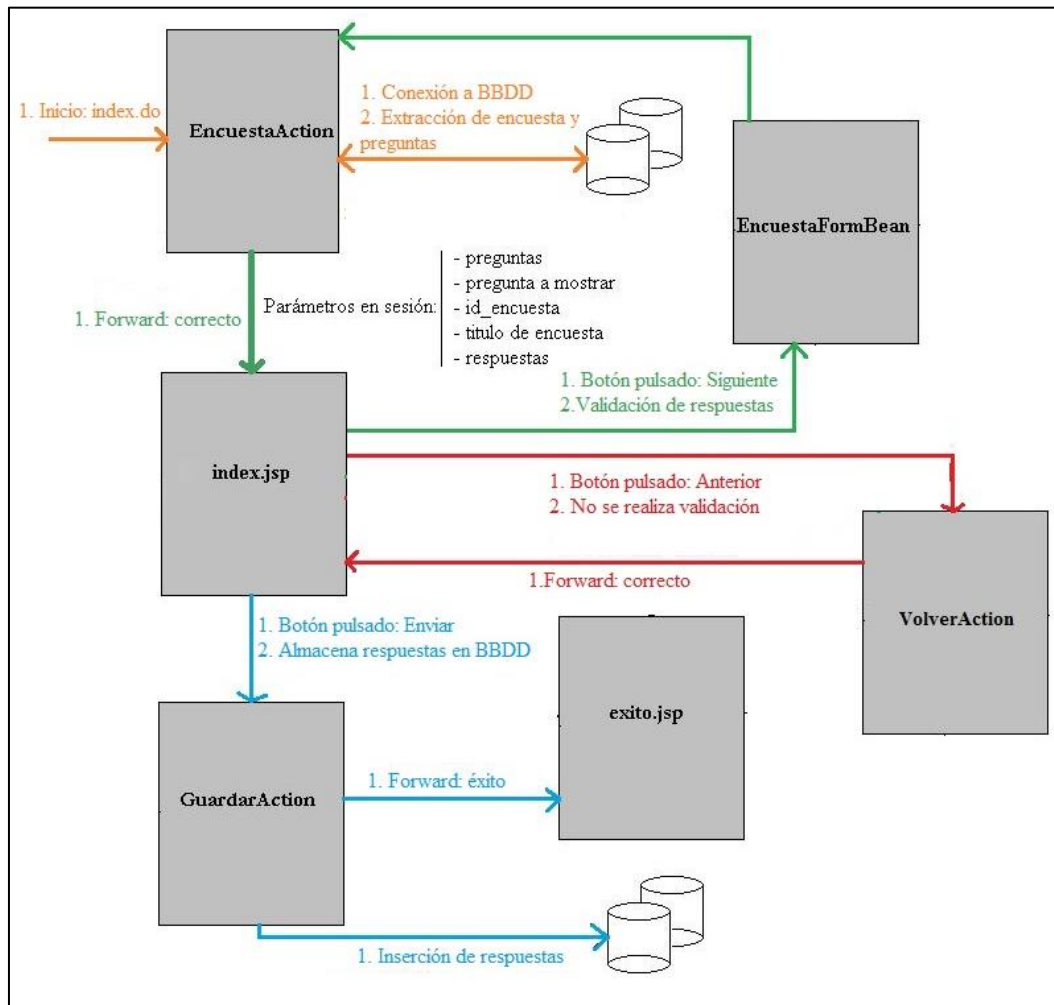
### **Generación y control de cuestionarios**

La generación y control de cuestionarios se realiza implementando los componentes ActionServlet, Actions y objetos pertenecientes, Java Beans, al modelo de la herramienta (DAOs, Vos y ActionForms), todos ellos mostrados en la Figura 11

#### **ActionServlet**

Este elemento componente únicamente aparece en el fichero web.xml. Actúa como un servlet con la diferencia de que su configuración la va a llevar el fichero struts-config.xml tal y como se le asigna en la configuración en el fichero web.xml. Redirige las peticiones que terminen en .do (\*do). Toda petición que recoja este servlet la vuelve a redirigir el fichero struts-config.xml a su correspondiente *Action*.

En la Figura 12 se muestra la conexión y el recorrido que se realiza en la aplicación entre componentes Action, JSP y ActionForm. A continuación se detallan todos estos componentes y se especifican de qué manera se conectan unos con otros en base a lo mostrado en la figura anterior.



**Figura 12. Representación de la comunicación entre componentes de Struts de la herramienta**

## Contenedor de Actions

Cada elemento *Action* tiene su configuración en el fichero *struts-config.xml* como se ha mencionado en el párrafo anterior. Este proyecto consta de 3 componentes de esta clase:

### *EncuestaAction*

Este componente tiene 2 casos en los que actúa de diferente manera.

El primer caso se da al iniciar la encuesta, cuando el componente que hereda de *ActionForm* no ha sido inicializado ya que aún no se ha realizado ninguna validación. Comprueba que el identificador de encuesta que viene como parámetro en la URL no este vacío ni nulo, sea mayor que 0 y tenga formato numérico. Realiza una conexión a la base de datos para extraer la encuesta según el identificador, recogido de la URL, e idioma actual de la Web, recogido también como parámetro 'L' de la URL y comprueba que exista alguna. En caso de incumplir alguna de las comprobaciones declaradas o no existir una encuesta con tal identificador, este componente devuelve un mensaje de error especificándolo. Una vez se posee la encuesta, abre

una nueva conexión a la base de datos para construir la estructura de preguntas. Por último, inicializa los siguientes parámetros en sesión que se utilizan a lo largo de la encuesta.

- título: título de la encuesta.
- preguntas: las preguntas de la encuesta.
- pregunta: pregunta a mostrar a continuación.
- respuestas: respuestas que han sido realizadas hasta el momento.
- id\_encuesta: identificador de la encuesta.
- flujo: preguntas por las que se ha pasado hasta el momento.
- contador
- preg\_restantes: preguntas restantes en el caso máximo.

Una vez ha pasado la primera pregunta y pasa por el validador, cuando llegue a este componente el *ActionForm* ya está inicializado y por tanto entra en el segundo caso. En este momento toma el array de respuestas de la pregunta actual, creado por el validador del formulario, y realiza la actualización y/o mantenimiento de respuestas y flujo, y la asignación de pregunta siguiente actualizando sus parámetros de sesión, proceso que se detalla en el apartado 5.2 Flujo adaptativo de preguntas en un cuestionario. También añade a un nuevo parámetro de sesión, llamado 'respuestas\_anteriores', las respuestas de la pregunta siguiente si las hubiese.

### ***VolverAction***

Este componente recupera los parámetros de sesión preguntas, flujo y respuestas. Con el parámetro 'flujo', que contiene un array con aquellas preguntas que ya han sido respondidas o por las que ya se ha navegado, recupera la pregunta anterior a la actual, que se corresponde con la inmediatamente anterior en dicho array. Con el identificador de la pregunta anterior ya establecida recupera sus respuestas si las hubiese, y actualiza el parámetro de sesión 'respuestas\_anteriores'. Actualiza también los parámetros 'contador' y 'preg\_restantes'.

### ***GuardarAction***

Este componente abre una conexión a base de datos para insertar todas las respuestas. En esta conexión recoge el siguiente valor de la secuencia SQL que aporta el identificador de usuario encuestado y ejecuta un procedimiento para insertar las respuestas. Por último resetea las respuestas en su parámetro de sesión.

### **Contenedor de JavaBeans**

El contenedor de JavaBeans contiene elementos de tipo VO (Value Object), DAO (Data Access Object) y *ActionForm*.

### ***Elementos ActionForm***

- EncuestaFormBean: Este componente se encarga de comprobar las respuestas tras presionar el botón 'Siguiente' en una pregunta (cuando se pulsa 'Anterior' no se realiza validación).



Valida las respuestas comprobando si la pregunta contiene subpreguntas o si es una sola pregunta.

Construye un objeto *RespuestaVO* por cada respuesta insertada. En caso de que no se responda a la pregunta crea una respuesta con el identificador de opción igual a '-1'. Esos objetos de tipo *RespuestaVO* los almacena en una array como atributo de *EncuestaFormBean*. Control específico para el tipo de respuesta 'otras' almacenando el texto insertado en un cuadro de texto.

### ***Elementos VO***

- **EncuestaVO**: Objeto que representa una encuesta. Contiene los atributos identificador, título e identificador de primera pregunta.
- **RegistroVO**: Objeto que se utiliza de manera intermedia en la creación de la estructura de preguntas en *PreguntaDAO*. Representa un registro del cursor de salida de los procedimientos almacenados para obtener preguntas y subpreguntas a los que se llama desde dicho objeto de tipo DAO. Sus atributos son 'id', 'texto', 'tipo', 'siguiente', 'id\_opcion', 'opcion', 'opcion\_orden', 'padre', 'suborden', 'dependiente' y 'opcion\_condicion'.
- **PreguntaVO**: Objeto clave en la implementación de la aplicación. Sus atributos son su identificador, su texto, el identificador de la pregunta padre (si la tuviese), una lista de opciones de tipo *OpcionVO*, el identificador del tipo de pregunta, el identificador de la pregunta siguiente, una lista de subpreguntas de tipo *PreguntaVO* y una estructura de tipo Map, que más tarde se especializa como *HashMap*, que representa sus flujos alternativos.
- **OpcionVO**: Objeto que se incluye en la lista de opciones de cualquier pregunta. Sus atributos son su identificador y su texto.
- **PatronVO**: Objeto que representa una encuesta acorde a un patrón y sus fechas de inicio y fin. Este objeto es tratado únicamente en la parte de la aplicación referente a la petición al servlet de una encuesta acorde al patrón de la página, con respuesta en formato JSON.
- **RespuestaVO**: Objeto que representa una respuesta a una pregunta y cuyos valores se insertan en la base de datos al finalizar la encuesta. Sus atributos son los identificadores de la pregunta y la opción, y el texto de la opción. El texto de la opción es el propio texto de esta en la base de datos a menos que sea la opción 'Otras' en cuyo caso se inserta el texto enviado por el usuario.

### ***Elementos DAO***

- **EncuestaDAO**: Contiene la llamada al procedimiento que extrae el título e identificadores de encuesta y primera pregunta por identificador de encuesta e idioma, generando un objeto de tipo *EncuestaVO*.
- **PreguntaDAO**: Implementa un método para extraer las preguntas de la encuesta ya estructuradas. Este método ejecuta dos procedimientos almacenados, de los cuales el

primero de ellos extrae las preguntas y el segundo extrae las subpreguntas. Para la salida de ambos procedimientos se generan objetos de tipo RegistroVO ya que aún no se tiene la estructura de preguntas completa. Una vez se tienen dos arrays de tipo RegistroVO, que contienen todas las preguntas y subpreguntas, se crea dicha estructura, para lo que se declara en este método un array de tipo PreguntaVO.

Para explicar el formato de los objetos RegistroVO se toma como ejemplo la Figura 13, donde se tiene una pregunta con identificador ‘1’ y otra con identificador ‘2’ que es dependiente de la primera. Para cada pregunta se devuelven tantos registros como opciones tenga, y en caso de que una pregunta sea dependiente de otra cada registro de cada opción, se repite tantas veces como opciones condicionantes haya. Las opciones condicionantes, columna ‘CONDICION’, son las opciones de la pregunta indicada por la columna ‘DEPENDIENTE\_DE’ que provocan que se vaya a esta pregunta, columna ‘ID’. Siguiendo el ejemplo de la imagen, el registro de la pregunta 2 con opción 7 se repite 2 veces, la primera para la opción condicionante 1 y la segunda para la opción condicionante 2. Esto representa que cuando en la pregunta 1 se selecciona la opción 1 o 2, va a la pregunta 2 y si no, sigue el flujo normal, indicado por la columna ‘SIGUIENTE’, yendo a la pregunta 3.

Con el identificador de cada registro se comprueba si en el array de preguntas existe alguna con dicho identificador y, si no existe, se crea la pregunta y se añade. Si la pregunta es dependiente de la respuesta de otra, columna ‘DEPENDIENTE\_DE’, se busca la pregunta de la que depende y a esta última se le añade en su tabla hash de flujos un nuevo par clave-valor cuya clave es la opción condicionante, y el valor la pregunta destino, columna ‘ID’. Para agregar las opciones a la pregunta se comprueba que esta ya no la tenga ya si una pregunta depende de otra por más de una condición.

ID	TITULO	TIPO	TIPO_TEXTO	SIGUIENTE	ID_OPCION	OPCION
1	¿Considera útil la inf	2	RADIO2	3	1	1
1	¿Considera útil la inf	2	RADIO2	3	2	2
1	¿Considera útil la inf	2	RADIO2	3	3	3
1	¿Considera útil la inf	2	RADIO2	3	4	4
1	¿Considera útil la inf	2	RADIO2	3	5	5
2	¿Por qué considera	1	RADIO	3	7	No he encontrado la información que buscaba
2	¿Por qué considera	1	RADIO	3	7	No he encontrado la información que buscaba
2	¿Por qué considera	1	RADIO	3	8	Contiene información relevante para mí, pero el nivel de detalle no es suf
2	¿Por qué considera	1	RADIO	3	8	Contiene información relevante para mí, pero el nivel de detalle no es suf
2	¿Por qué considera	1	RADIO	3	9	Otras respuestas: (indique cuáles)
2	¿Por qué considera	1	RADIO	3	9	Otras respuestas: (indique cuáles)
	OPCION_OTROS	OPCION_TABULADA	OPCION_ORDEN	PADRE	SUB_ORDEN	DEPENDIENTE_DE
	0	0	1			
	0	0	2			
	0	0	3			
	0	0	4			
	0	0	5			
	0	0	1			1
	0	0	1			1
	0	0	2			1
	0	0	2			1
	1	0	3			1
	1	0	3			1

**Figura 13. Cursor de salida del procedimiento almacenado que extrae preguntas**

Tras almacenar las preguntas padre (preguntas que no son subpreguntas), se toma el array de subpreguntas resultantes de la consulta a base de datos cuyo resultado se muestra en la Figura 14. Para incluirlas en la estructura de preguntas creada hasta el momento se realizan los siguientes pasos:

a) Buscar su pregunta padre

b) Añadirla como subpregunta a la pregunta padre en caso de que no esté añadida y añadirle la opción. En caso de que ya esté añadida simplemente se añade la opción.

ID	TITULO	TIPO	TIPO_TEXTO	SIGUIENTE	ID_OPCION	OPCION
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		1	1
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		2	2
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		3	3
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		4	4
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		5	5
4	<strong>Información básica acerca del Sistema Estadístico Euro	2	RADIO2		6	No sabe / No contesta
OPCION_OTROS	OPCION_TABULADA	OPCION_ORDEN	PADRE	SUB_ORDEN	DEPENDIENTE_DE	CONDICION
0	0	1	3	1		
0	0	2	3	1		
0	0	3	3	1		
0	0	4	3	1		
0	0	5	3	1		
0	0	6	3	1		

*Figura 14. Cursor de salida del procedimiento almacenado que extrae subpreguntas*

- **RespuestaDAO:** Implementa un método que realiza una conexión a base de datos para, primero extraer el identificador de encuestado según la secuencia SQL que lo controla y a continuación, insertar las respuestas de la encuesta asociadas al identificador de encuestado obtenido por la secuencia.

Este módulo es sensible a Inyección SQL (SQL Injection) dado que va a ejecutar texto insertado por el usuario en una consulta. Este peligro se previene utilizando PreparedStatement, de manera que se parametriza la query que se va a ejecutar.

- **PatronDAO:** Implementa un método que realiza una conexión a base de datos para ejecutar un procedimiento que devuelve todos los patrones y construye objetos PatronVO.

### 4.2.3. Interfaz gráfica

La interfaz gráfica de la herramienta se implementa en dos bloques: el cuadro de diálogo que se muestra tras encontrar el cuestionario apropiado, y la presentación de la encuesta. Ambos bloques se detallan a continuación.

#### **Cuadro de diálogo para confirmación**

Esta aplicación presenta al usuario un cuadro de diálogo para informarle de que hay una encuesta disponible, que quizá estaría interesado en completar, una vez se ha comprobado que hay una encuesta disponible para la sección y que la cookie tiene valor '0' o bien no existe.

Tras dichas comprobaciones se procede a incluir en la página la librería *jQuery-ui* perteneciente a *jQuery*<sup>16</sup> que se utiliza para dar un formato visual al cuadro. ¿Por qué incluir la librería en este momento? El motivo de adjuntar esta librería en este lugar es evitar la sobrecarga innecesaria a la hora de descargar del lado del servidor el contenido de la página ya que esta librería es propia de este aplicativo y no es común al resto de contenidos del sitio Web.

<sup>16</sup> jQuery es una biblioteca de javascript cuyo código es libre y abierto y que aporta funcionalidad para la creación de interfaces, manejo de eventos, animaciones y el uso de Ajax. jQuery.UI aporta funcionalidad específica en la creación de efectos visuales, interacciones y widgets.

Además del fichero de extensión ‘.js’, que contiene las funciones javascript de *jQuery-ui*, se incluye también el fichero de extensión ‘.css’ que aporta el estilo a la interfaz creada con esta librería.

En la Figura 1 se muestra como es el cuadro de diálogo utilizado en esta herramienta cuando se pregunta al usuario si desea realizar una encuesta disponible en esa página/sección de la Web del INE con tres opciones, que son “Realizar”, “Más tarde” o “Ya la hice”.

### **Presentación de encuesta**

La interfaz gráfica de las páginas que se muestran a lo largo de la encuesta se compone de tres partes: cabecera, cuerpo y pie.

La cabecera y el pie son ficheros de extensión .jsp que se incluyen con la etiqueta JSTL ‘<c:import>’ en las páginas .jsp del proyecto, que componen el cuerpo de la interfaz.

Para dar formato a la interfaz del cuerpo se ha creado un fichero de extensión ‘.css’ que contiene el estilo que se ha dado a botones, títulos, preguntas, opciones, cuadros de texto en cuanto a tamaño, color, fondo, etc. Este fichero tiene que seguir las pautas establecidas por el INE para mantener toda la página Web y sus aplicativos con el mismo estilo. Para cumplir la normativa interna, el fichero ‘.css’ propio de la aplicación utiliza los mismos códigos de colores, tamaños de letra y tipos de letra que contienen los ficheros ‘.css’ generales pero especializando los elementos html y sus clases de la encuesta.

## **5. Gestión de cuestionarios**

La gestión de cuestionarios se apoya en un esquema de base de datos cuyos apartados se definen en el apartado *Estructuras de datos*. Además se detalla la lógica que implementada para conseguir un flujo adaptativo de preguntas en el apartado *Flujo adaptativo de preguntas en un cuestionario*, y por último se explica, en el apartado *Búsqueda y selección de cuestionarios*, el método para obtener un cuestionario a partir de la URL de una página del sitio Web.

### **5.1. Estructuras de datos**

La encuesta se forma a partir del diseño del esquema de base de datos. En este apartado se explican todas las tablas del esquema creado en pequeños grupos que representan bloques de la encuesta. Estos bloques son las relaciones *Pregunta-Subpregunta*, *Encuesta-Pregunta-Tipo*, *Pregunta-Flujo*, *Pregunta-Opcion*, *Encuestas-Respuestas* y *Encuestas-Patronos*.

Por cada relación se mostrará el diagrama ER de las tablas que intervienen en ella, donde se pueden ver los campos, claves primarias y claves foráneas de cada tabla. En las tablas las llaves amarillas representan las claves primarias, las llaves azules las claves foráneas, el tipo de dato viene indicado en color azul tras el nombre de la columna, el símbolo de prohibido indica que el campo no puede ser nulo y tras él se indica si pertenece a algún índice.

#### **Relación Pregunta-Subpregunta**

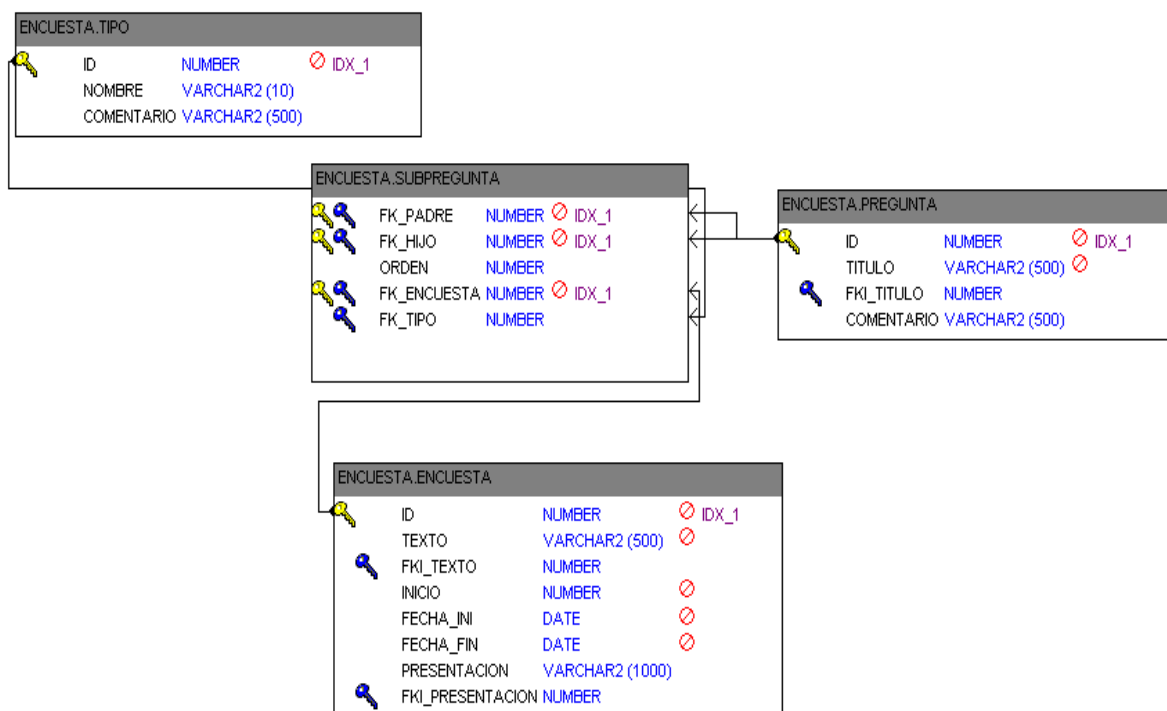
El esquema diseñado para el aplicativo soporta múltiples encuestas diferentes con múltiples preguntas cada una de ellas. Estas preguntas pueden ser de dos clases:

- **Simples:** tienen opciones y tipo.
- **Compuestas:** contienen preguntas simples por lo que directamente ellas no tienen ni tipo ni opciones si no las preguntas que contiene. También se hace referencia a esta clase de pregunta como pregunta padre. Aunque las preguntas compuestas o preguntas padre no tengan tipo, se les asigna uno ya que en caso de que la subpregunta tenga tipo nulo, adoptará el tipo de la pregunta padre

La relación entre preguntas compuestas y sus subpreguntas se resuelve en la tabla SUBPREGUNTA. En la Figura 15 se muestra la relación entre esta tabla y las tablas PREGUNTA, TIPO y ENCUESTA. El campo *presentacion* de la tabla ENCUESTA indica el contenido de la página de presentación de la encuesta. En caso de ser nulo no se muestra **página de presentación**. La columna *fk\_padre* indica la pregunta a la que pertenece y junto con el tipo asociado aporta a la subpregunta independencia del tipo respecto de la pregunta padre.

Las preguntas que únicamente sean dependientes de otra pregunta sólo aparecen en la tabla SUBPREGUNTA. En el siguiente punto se detalla la tabla que contiene las preguntas compuestas por subpreguntas.

Además, cada subpregunta podrá tener un tipo asociado para cada pregunta padre en la que aparezca además para cada encuesta. En caso de que dicho campo venga como nulo, tomará el tipo asociado a la pregunta padre.



*Figura 15. Diagrama ER de la relación entre preguntas, encuestas y subpreguntas*

### **Relación Encuesta - Pregunta - Tipo**

La relación entre la encuesta y la pregunta es una relación *N:1* tal que existen N preguntas por cada encuesta, pero se diseña con una relación *N:N* de manera que cualquier pregunta se pueda

Por tanto, la relación entre preguntas y encuestas, cuyo diagrama ER se muestra en la Figura 16, se establece mediante la tabla *ENC\_PREG* que contiene claves foráneas a las claves primarias de *ENCUESTA* y *PREGUNTA* además de una columna de nombre *fk\_siguiente*, que indica el orden natural a seguir en la navegación entre preguntas. Por orden natural se entiende que no se aplica ningún flujo extraordinario. La clave foránea de la pregunta siguiente puede ser nula y por tanto no forma parte de la clave ya que no puede haber más de una pregunta siguiente para una pregunta. Un registro con identificador de pregunta siguiente nulo es indicativo de que esa pregunta finaliza la encuesta ya que no hay ninguna pregunta tras ella.

```
graph TD
    subgraph ENCUESTA_TIPO [ENCUESTA.TIPO]
        ID_TIPO((ID))
        NOMBRE_TIPO[NOMBRE]
    end
    subgraph ENCUESTA_ENCUESTA [ENCUESTA.ENCUESTA]
        ID_ENCUESTA((ID))
        TEXTO_ENCUESTA[TEXTO]
        FK1_TEXTO((FK1_TEXTO))
        INICIO_ENCUESTA[INICIO]
        FECHA_INI_ENCUESTA[FECHA_INI]
        FECHA_FIN_ENCUESTA[FECHA_FIN]
        PRESENTACION_ENCUESTA[PRESENTACION]
        FK1_PRESENTACION((FK1_PRESENTACION))
    end
    subgraph ENCUESTA_ENC_PREG [ENCUESTA.ENC_PREG]
        FK_ENCUESTA((FK_ENCUESTA))
        FK_PREGUNTA((FK_PREGUNTA))
        FK_SIGUIENTE((FK_SIGUIENTE))
        FK_TIPO((FK_TIPO))
    end
    subgraph ENCUESTA_PREGUNTA [ENCUESTA.PREGUNTA]
        ID_PREGUNTA((ID))
        TITULO_PREGUNTA[TITULO]
        FK1_TITULO((FK1_TITULO))
        COMENTARIO_PREGUNTA[COMENTARIO]
    end

    ID_TIPO --> FK1_TEXTO
    ID_TIPO --> FK1_PRESENTACION
    ID_TIPO --> FK1_PRESENTACION
    ID_ENCUESTA --> FK_ENCUESTA
    ID_ENCUESTA --> FK_PREGUNTA
    ID_ENCUESTA --> FK_SIGUIENTE
    ID_ENCUESTA --> FK_TIPO
    ID_PREGUNTA --> FK1_TITULO
    ID_PREGUNTA --> COMENTARIO_PREGUNTA
```

De la misma manera que una pregunta puede pertenecer a diferentes encuestas, también puede tener un tipo diferente según la encuesta por lo habrá una relación N:N entre la encuesta, la pregunta y el tipo de pregunta. Para la búsqueda sobre esta tabla la clave es la unión de las tres claves foráneas a las claves primarias de las tres tablas implicadas.

30

Las encuestas no van a tener una vida ilimitada pero sus respuestas deben permanecer en la base de datos, ya que cualquier estudio sobre ella se realiza tras finalizar el período de muestreo. Para solucionar esa necesidad se aplica una duración a cada encuesta indicada por su inicio y su fecha de caducidad. Dichos atributos son importantes a la hora de seleccionar la encuesta que se debe realizar al usuario y permiten cargar los datos de una encuesta sin obligar al aplicativo a que muestre la encuesta desde ese mismo instante.

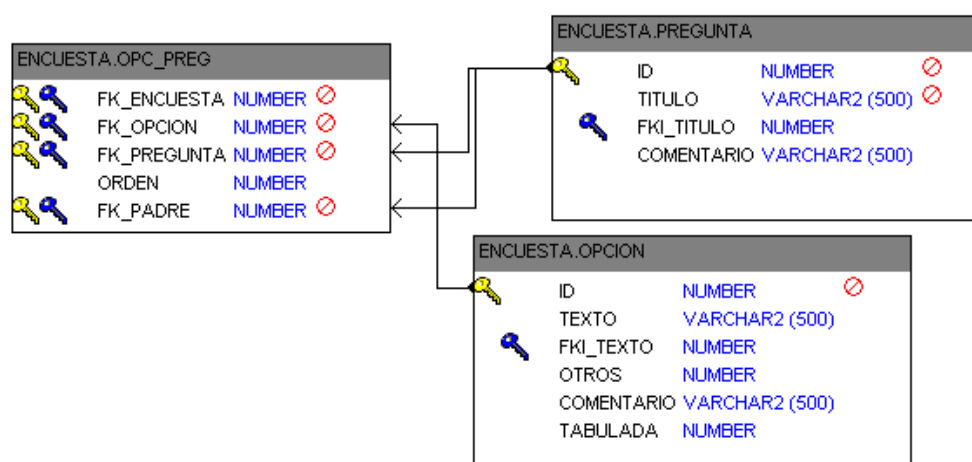
Por cada encuesta se indica el identificador de la primera pregunta a mostrar. Esta solución se implementa al ser la opción óptima en cuanto a rendimiento y velocidad de búsqueda tras la alternativa inicial de incluir por cada pregunta un identificador de orden en la encuesta, la cual quedó inútil al aportar flujo alternativo a las encuestas.

### ¿Por qué no referir al tipo de subpregunta en la tabla ENC\_PREG?

En la tabla *ENC\_PREG* únicamente se incluyen las preguntas padre y preguntas sin subpreguntas, luego no se puede hacer referencia al tipo de subpregunta. Las subpreguntas llevan asociado su tipo en la tabla *SUBPREGUNTA* y aquellas subpreguntas que tengan tipo asociado nulo, tomarán el tipo asociado a la pregunta padre. De esta manera se permite a un grupo de subpreguntas de la misma pregunta padre tener diferentes tipos.

### Relación Pregunta-Opción

Las opciones se tratan como un objeto único y no como bloques asignados a una pregunta. Este tratamiento permite optimizar la cantidad de opciones almacenada de la misma forma que se tratan las preguntas. Una opción puede pertenecer a múltiples preguntas y múltiples encuestas, incluso a varias preguntas diferentes dentro de la misma encuesta. Para poder tener esta relación entre las preguntas, las opciones y encuestas se dispone de una estructura intermedia entre los tres elementos a relacionar.



*Figura 17. Diagrama ER de la relación entre encuestas, preguntas y opciones*

Se puede incluir el identificador de opción en la tabla que relaciona encuesta con pregunta, y repetir un registro ‘n’ veces siendo ‘n’ el número de opciones que tuviese, manteniendo fijos los identificadores de encuesta, pregunta y pregunta siguiente, pero se toma la decisión de diseñar

otra tabla en la que se incluyan los identificadores de pregunta, encuesta, opción y además un atributo que nos indique el orden de esa opción en la pregunta.

### ¿Por qué no referir a las opciones en la tabla ENC\_PREG?

No existe la posibilidad de añadir la relación entre pregunta y opciones en dicha tabla, ya que la opción debe formar parte de la clave primaria ya que si no sólo se puede añadir un registro con una única opción por pregunta. Al hacer que sea clave primaria provoca que no pueda ser nula y aquí se encuentra otro problema. Las preguntas complejas, de la misma manera que no tienen tipo, no tienen opciones y al no poder ser nulo el identificador de opción hay que asignarle alguna a la pregunta obligatoriamente.

### Relaciones Pregunta-Flujo

La parte más compleja del diseño de la base de datos es prepararla para la posible existencia de encuestas con orden entre preguntas no natural. Por orden no natural se entiende que no tiene por qué seguirse el orden natural indicado en el identificador mencionado si no que pueden existir una o varias respuestas que actúen como condicionante en el camino seguir entre preguntas.

Como se ha mencionado en la relación entre preguntas y encuestas, en la tabla *ENC\_PREG* se utiliza un identificador *fk\_siguiente* que es clave foránea de la pregunta que va a continuación siguiendo un orden natural en el que no interviene ningún condicionante.

Para resolver los casos de orden no natural o flujo alternativo se utiliza la tabla *FLUJO*. Esta tabla contiene registros con los identificadores de pregunta, encuesta a la que pertenece, opción que representa la condición y pregunta a la que se va a navegar a continuación. En la Figura 18 se muestran las relaciones de la tabla *FLUJO* con las tablas *PREGUNTA*, *OPCION* y *ENCUESTA*.

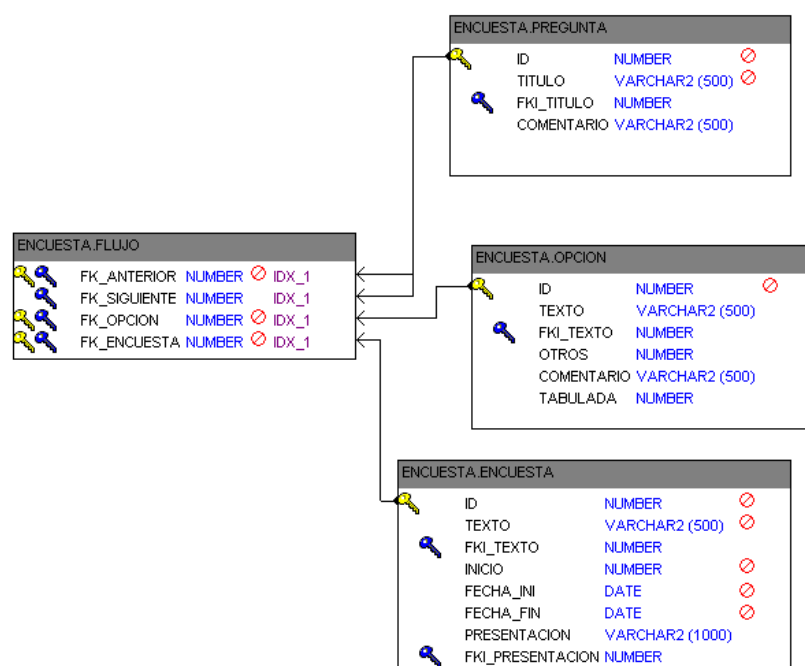


Figura 18. Diagrama ER de la relación entre encuestas, preguntas y flujo



La clave primaria de esta tabla la forman los identificadores de la pregunta anterior, la encuesta y la opción condicionante. El resultado de esa clave primaria es el identificador de la pregunta siguiente. La clave foránea de la pregunta siguiente no puede formar parte de la clave ya que no puede haber más de una pregunta siguiente para la misma opción de una pregunta. De la misma forma que en la tabla de flujo natural un identificador de pregunta siguiente es indicativo de que es la última pregunta, se permite el caso de que dependiendo de la respuesta de una pregunta se provoque un final de encuesta anticipado al esperado según el orden natural.

### Relación Encuesta-Respuestas

Para almacenar las respuestas de los usuarios en base de datos se crea la tabla *RESPUESTA*. Esta tabla contiene los identificadores de la encuesta, pregunta y opción, tablas con las que se relaciona tal y como muestra la Figura 19, además de un identificador numérico único de usuario encuestado que se autogenera gracias a una secuencia creada para dicho campo. Por último se almacena el texto de la opción para aquellas preguntas cuya respuesta pueda ser libre de redactar por el usuario.

Dada la clave primaria, formada por los 4 identificadores, se permite la introducción de múltiples respuestas por pregunta lo que en algunos casos es correcto, por ejemplo preguntas de tipo ‘checkbox’, pero incorrecto en otras como las de tipo ‘radio’. En el módulo controlador de la aplicación se realiza esta comprobación tal y como se detalla en el apartado ‘Flujo adaptativo de preguntas en un cuestionario’ (5.2).

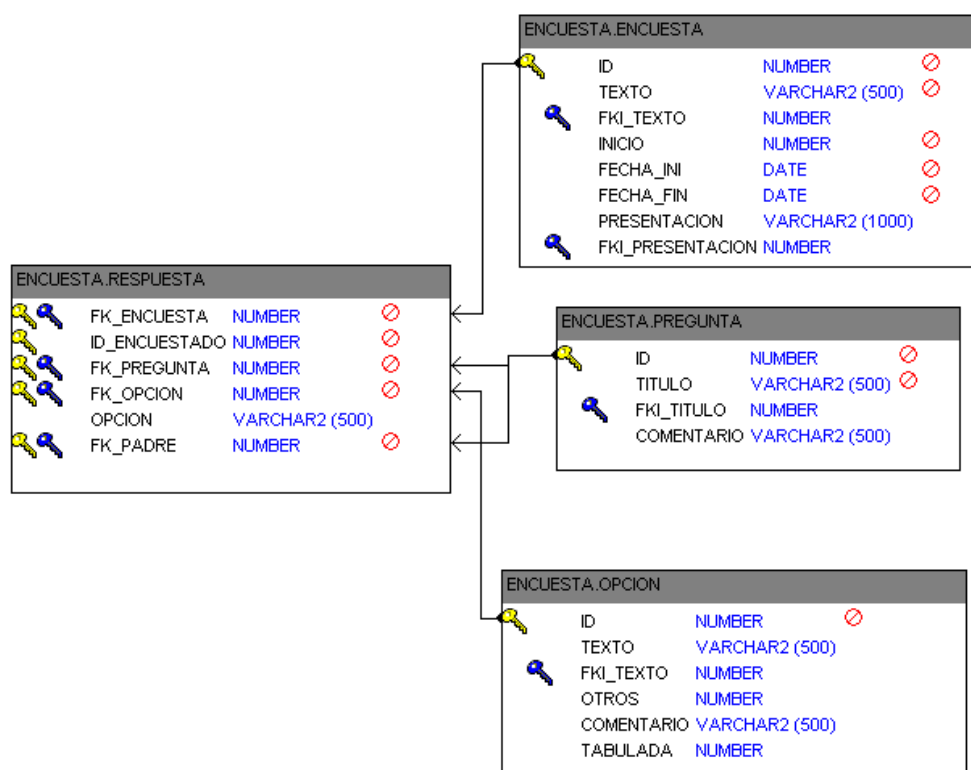


Figura 19. Diagrama ER de la relación entre respuestas, preguntas, opciones y encuestas

## Relación Encuestas-Patrones

Para la fase de la aplicación que designa una encuesta a la zona de la página Web en la que está situado el usuario, se diseña una tabla de nombre *PATRONES* que se relacione, como se puede observar en la Figura 20, con la de encuestas (tabla *ENCUESTA*) mediante una clave foránea al identificador de encuesta.

La tabla *PATRONES* aporta para cada encuesta uno o más patrones. Estos patrones, o expresiones regulares, definen una regla que va a describir la página o sección de la Web donde el usuario está situado de manera que las diferencien de las demás.

Aunque el usuario sólo puede realizar una encuesta como máximo en la página/sección en la que se encuentre, su patrón puede tener asociadas diferentes encuestas y una encuesta puede tener asociadas múltiples patrones. Por esta razón la clave primaria de la tabla *PATRONES* la forman el identificador de encuesta, que es clave foránea de la tabla *ENCUESTA*, y el patrón.

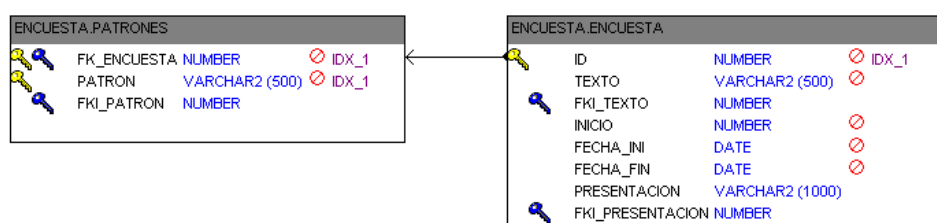


Figura 20. Diagrama ER de la relación entre encuestas y patrones de páginas.

## 5.2. Flujo adaptativo de preguntas en un cuestionario

La principal aportación de la herramienta es la capacidad para controlar un flujo adaptativo de preguntas y respuestas. En el apartado anterior, *Estructuras de datos*, se explica cómo se prepara el flujo alternativo en la base de datos pero, ¿cómo se controla desde el servidor de aplicaciones? En este apartado se detalla cómo se controla la navegación entre preguntas y el mantenimiento de respuestas.

### 5.2.1. Navegación entre preguntas

Los elementos del gráfico adjuntado en el apartado 4.2.2 *Gestión de cuestionarios* que toman parte en el control de navegación entre preguntas son los *Action* EncuestaAction y VolverAction, el ActionForm EncuestaFormBean y la página de interfaz *index.jsp*.

Los elementos *Action* son los encargados de establecer la siguiente pregunta a mostrar, cada uno de ellos siguiendo un algoritmo.

EncuestaAction se invoca cuando el usuario pulsa el botón “Siguiente” en la interfaz gráfica. Debido a la configuración de la aplicación diseñada en el fichero *struts.xml* (3.1.3 Apache Struts II), antes de pasar por EncuestaAction, EncuestaFormBean valida el formulario.

Una vez validado el formulario comienza el siguiente procedimiento para obtener cuál es la siguiente pregunta:

1. El caso de prioridad máxima a evaluar es que la pregunta actual ya estuviese respondida y que la respuesta actual no varíe. En este caso, la pregunta siguiente la recupera del array de preguntas ya realizadas que hay en sesión. Esta pregunta es la que esté en la siguiente posición del array a la que esté la pregunta actual.
2. ¿Tiene flujo alternativo la pregunta actual? Hay que recordar que, según el diseño de la base de datos, el flujo alternativo tiene mayor prioridad sobre el orden natural luego es lo primero que hay que comprobar. Si la tabla hash de flujos que tiene la pregunta contiene una clave que se corresponda con el identificador de la respuesta seleccionada, el identificador de la pregunta siguiente es el valor para esa clave en la tabla hash.
3. En caso de que no exista ninguna entrada en la tabla con clave igual al identificador de la respuesta, se selecciona como pregunta siguiente aquella cuyo identificador se corresponde con el indicado por la pregunta actual siguiendo el orden natural.

Tanto para el caso 2 como el caso 3, EncuestaAction actualiza el array de preguntas ya mostradas en sesión añadiendo la pregunta actual. Si elige pregunta siguiente por el caso 1 no la añade ya que la duplica.

Para el caso de preguntas con subpreguntas primero construye una array auxiliar de respuestas con las respuestas seleccionadas (identificador de opción igual a -1 si no ha respondido).

Tras tener el array auxiliar comprueba si existían o no respuesta a estas subpreguntas.

- Respuestas no existentes: añade las respuestas del array auxiliar. Calcula la pregunta siguiente por flujo normal.
- Respuestas existentes: Comprueba si cambian. Si cambian, actualiza el flujo de preguntas recorridas y el array de respuestas de la encuesta. Calcula la pregunta siguiente por el flujo de preguntas recorridas. Si no lo hay recurre al flujo normal de la pregunta padre.

VolverAction se ejecuta cuando el usuario pulsa el botón “Anterior” en la página mostrada. Para calcular cuál es la pregunta previa a la actual recurre al array de preguntas que permanece en sesión con las preguntas que han sido mostradas. Antes de terminar su ejecución y avanzar a la ejecución del fichero de interfaz, actualiza también los parámetros de sesión que almacenan la pregunta a mostrar y el array de preguntas mostradas. No va a eliminar la pregunta actual del array de preguntas mostradas ya que este array se utiliza en EncuestaAction cuando vuelva a ir hacia adelante tras haber navegado hacia atrás ahorrándose así la comprobación de flujo alternativo o natural.

### **¿Por qué no mirar que pregunta de la encuesta tiene asociado como pregunta siguiente a la actual?**

No puede seguir ese criterio ya que, según el esquema diseñado en la base de datos y por la propia lógica de mantener un flujo adaptativo, pueden darse múltiples preguntas que tuviesen como pregunta siguiente a la actual, llevando por lo tanto a una confusión y mala interpretación del concepto de pregunta a anterior en un flujo adaptativo.

### 5.2.2. Mantenimiento de respuestas

Ofrecer un flujo adaptativo no es únicamente poder optar a varias preguntas desde una posición de la encuesta, sino que también conlleva mantener y actualizar las respuestas de cada una de las preguntas. Para mantener las respuestas a lo largo de la aplicación se utiliza la sesión, donde se almacena un array de respuestas.

VolverAction es el módulo que se encarga de realizar la tarea de recuperación de respuestas de una pregunta cuando se navega hacia atrás. Este módulo, invocado al pulsar el botón anterior en el formulario, establece la pregunta a mostrar y una vez hecho este proceso, recupera la/s respuesta/s que tenga asociada/s dicha pregunta según su identificador en el array de respuestas almacenado en la sesión.

En esta aplicación las respuestas se almacenan en la base de datos en la tabla *RESPUESTA* cuya estructura se muestra en la Figura 19. De la manera en que está construida dicha tabla, se permite insertar varias respuestas por pregunta por motivo de las preguntas con múltiple respuesta, de tipo *checkbox*, por lo que si no se controlase la actualización de respuestas, la base de datos acepta la inserción de preguntas que, según su tipo, aceptan una única respuesta.

Ante la incapacidad por parte de la base de datos de controlarlo, EncuestaAction actualiza las respuestas cada vez que se navegue por la encuesta en ambos sentidos.

#### **¿Qué criterio sigue EncuestaAction al actualizar las respuestas?**

EncuestaAction interviene cuando el flujo seguido es hacia adelante por lo que hay que obtener la/s respuesta/s seleccionada/s. A continuación se comprueba si ya existe en el array de respuestas alguna para la pregunta actual. En caso de que no exista simplemente se añade. En caso contrario, se compara la respuesta actual y si coincide con la existente no se actualiza el array de respuestas, mientras que si la respuesta ha variado se elimina del array la anterior para añadir la nueva respuesta a la pregunta.

Si la respuesta a la pregunta actual ya existía y no ha variado, significa que el usuario ha ido navegando hacia atrás y ahora vuelve hacia adelante luego EncuestaAction, en este caso, también tiene que recuperar las respuestas de la pregunta siguiente a partir del array de respuestas mantenido en sesión y el identificador de la pregunta siguiente ya obtenida.

## 5.3. **Búsqueda y selección de cuestionarios**

Una importante aportación de la herramienta desarrollada es la búsqueda de encuestas apropiadas para la página en que este el usuario. El sitio Web del INE es de grandes dimensiones y tiene multitud de apartados de diferente temática, por lo que es vital que los cuestionarios se adecuen al contenido de la página en la que se encuentra el usuario.

### 5.3.1. Uso de patrones

Para aportar la selección de cuestionarios específicos a secciones, la aplicación se sirve de patrones. Para dar valor a un patrón se selecciona una parte de la URL de cada página que la distinga del resto del sitio Web donde no quiere verse dicho cuestionario.

Para la encuesta sobre información estadística europea, la primera encuesta que se pone en funcionamiento sobre esta plataforma, se toma como patrón el parámetro “p” de la URL, que es el más restrictivo de todos los posibles. Este parámetro es único para todas las páginas del sitio Web.

El valor de ese parámetro es un identificador numérico luego el patrón que se forma para dirigir este cuestionario a esta página es el siguiente:

(.)\*identificador(.)\*

### 5.3.2. Interpretación de peticiones de búsqueda

Al cargarse una página, por medio de Javascript se toma la URL de la misma. De esta URL se codifican los caracteres especiales tales como ‘/’, ‘?’, ‘:’, ‘@’, ‘&’, ‘=’, ‘+’, ‘\$’ o ‘#’. El componente servlet se encarga de pedir a la base de datos todos los patrones para los que la fecha actual cumpla su rango de fechas. Tras recibir una lista de patrones, el servlet aplica a la URL recibida como parámetro, previamente descodificada, la función de java “*public boolean matches(String expresion\_regular)*”, siendo “*expresion\_regular*” el patrón extraído de base de datos. Si la función devuelve *true* significa que ese patrón es válido y por lo tanto la encuesta asociada a este patrón es la adecuada para esta página Web.

El servlet devuelve los datos de la encuesta en formato JSON, los cuales parsea la misma función javascript getJSON. A partir de este momento propone la realización de la encuesta al usuario.

## 6. Conclusiones

En este proyecto se ha desarrollado una herramienta flexible para la creación, presentación y gestión de cuestionarios en línea que se usará en el Instituto Nacional de Estadística para la puesta en funcionamiento de cuestionarios y posterior recogida de respuestas.

La herramienta desarrollada ofrece a los usuarios del sitio web del INE distintos cuestionarios según la página en la que se encuentren, de manera que tengan conocimiento del tema sobre el que son preguntados.

La herramienta aporta como novedad la navegación entre preguntas y el flujo adaptativo en ambos sentidos, el mantenimiento de respuestas en sesión, y el análisis de patrones para obtener el cuestionario adecuado.

A lo largo de este documento se han detallado los métodos seguidos para implementar las aportaciones mencionadas, y las tecnologías utilizadas para el desarrollo de la herramienta.

La herramienta no está cerrada por completo ya que continúan surgiendo ideas que podrían dotarla de mayor funcionalidad, como que los usuarios reciban sugerencias de encuestas a realizar vía correo electrónico en lugar de mediante análisis de patrones, o aumentar el número de tipos de pregunta a medida aceptados a medida que se vayan solicitando cuestionarios.

## Anexo I. Ejemplos de encuestas generadas

En este anexo se muestra el seguimiento de un cuestionario de prueba implementado sobre la herramienta diseñada.

The screenshot shows the top navigation bar of the Instituto Nacional de Estadística (INE) website. The header includes the INE logo and a search bar. Below the header, there is a grid of service links: 'El INE', 'Métodos y estándares', 'Presupuesto', 'Censo electoral', 'INEbase', 'Formación y empleo', 'Productos y servicios', and 'Sede electrónica'. The main content area displays the survey title 'Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE' and a progress indicator 'Le quedan por responder como máximo 8 preguntas.' The first question is '1. ¿Considera útil la información de este apartado?' with a subtext 'Por favor, puntúe de 5 (muy útil) a 1 (nada útil).' and five radio button options labeled 1 through 5. A 'Siguiente' button is located at the bottom right of the question area. The footer contains copyright information '© INE 2013' and links for 'Accesibilidad', 'Aviso de seguridad', 'Aviso legal', 'Ayuda', and 'Dónde encontramos', along with logos for 'Año internacional de la Estadística', 'Sistema estadístico europeo', and 'Contactar INE'.

*Figura 21. Primera pregunta del cuestionario*

En la primera pregunta, mostrada en la Figura 21, se tienen 5 opciones, de las cuáles la 1 y la 2 tienen un flujo alternativo que lleva a una pregunta aclaratoria sobre el porqué de esta respuesta, Figura 22.

This screenshot shows the second question of the survey. The progress indicator now says 'Le quedan por responder como máximo 7 preguntas.' The question is '2. ¿Por qué considera poco o nada útil la información sobre el Sistema Estadístico Europeo de nuestra web?' It features three radio button options: 'No he encontrado la información que buscaba', 'Contiene información relevante para mí, pero el nivel de detalle no es suficiente', and 'Otras respuestas: (indique cuáles)'. The third option is selected, and a text input field labeled 'otras causas' is visible. 'Anterior' and 'Siguiente' buttons are positioned below the options. The footer is identical to the one in Figure 21.

*Figura 22. Segunda pregunta del cuestionario*

En esta pregunta se muestran 3 opciones, de las cuáles una de ellas se acompaña de un cuadro para que el usuario introduzca una respuesta abierta. Como se puede ver en las Figuras 21 y 22, encima de la pregunta se muestra el máximo de preguntas que quedan por responder, y el título de la pregunta viene acompañado por el número de pregunta en la que se encuentra el usuario.

Si en la Figura 21 se hubiese respondido la opción 3,4 o 5, la segunda pregunta que se representa es la indicada por el flujo normal, y que se muestra en la Figura 23.

**IN e Instituto Nacional de Estadística**

01 El INE 03 Métodos y estándares 05 Puestos 07 Censo electoral  
02 INEbase 04 Formaciones y empleos 06 Productos y servicios 08 Sede electrónica

Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE

Le quedan por responder como máximo 6 preguntas.

**2. Indique el nivel de relevancia (interesante para sus necesidades) de la información incluida en este apartado.**  
Por favor, puntúe de 5 (muy relevante) a 1 (nada relevante):

**Información básica acerca del Sistema Estadístico Europeo**  
☒ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ No sabe / No contesta

**Información sobre la calidad de las estadísticas europeas**  
☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ No sabe / No contesta

**Tablas estadísticas europeas más consultadas**  
☐ 1 ☐ 2 ☒ 3 ☐ 4 ☐ 5 ☐ No sabe / No contesta

**Enlaces a la web del Sistema Estadístico Europeo y a la de otras oficinas de estadística**  
☐ 1 ☐ 2 ☐ 3 ☒ 4 ☐ 5 ☐ No sabe / No contesta

[Anterior](#) [Siguiente](#)

© INE 2013 [Accesibilidad](#) [Aviso de seguridad](#) [Aviso legal](#) [Ayuda](#) [Dónde encontramos](#) [Año internacional de la Estadística](#) [Sistema estadístico europeo](#) [Contactar INE](#)

*Figura 23. Segunda pregunta siguiendo orden normal o tercera siguiendo el flujo adaptativo*

La pregunta representada en la Figura 23 contiene 4 subpreguntas con 6 opciones de tipo radio horizontal cada una de ellas.

Si se continúa desde la pregunta que muestra la Figura 22, pasando por la pregunta de la Figura 23, mostrada como tercera pregunta, se llega a la cuarta pregunta, que se muestra a continuación en la Figura 24.

**IN e Instituto Nacional de Estadística**

01 El INE 03 Métodos y estándares 05 Puestos 07 Censo electoral  
02 INEbase 04 Formaciones y empleos 06 Productos y servicios 08 Sede electrónica

Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE

Le quedan por responder como máximo 4 preguntas.

**4. Indique su nivel de confianza en las estadísticas producidas por el Sistema Estadístico Europeo.**  
Por favor, puntúe de 5 (mucha confianza) a 1 (nada de confianza):

☐ 1 ☒ 2 ☐ 3 ☐ 4 ☐ 5 ☐ No sabe / No contesta

[Anterior](#) [Siguiente](#)

© INE 2013 [Accesibilidad](#) [Aviso de seguridad](#) [Aviso legal](#) [Ayuda](#) [Dónde encontramos](#) [Año internacional de la Estadística](#) [Sistema estadístico europeo](#) [Contactar INE](#)

*Figura 24. Cuarta pregunta del cuestionario*

Tras esta pregunta se puede ir a la pregunta siguiente por flujo normal, opción no sabe/no contesta o por flujo alternativo, opciones 1 y 2 a una pregunta y opciones 3,4 y 5 a otra pregunta. Para mostrar un ejemplo se selecciona en la Figura 24 la opción 2, y la pregunta siguiente es la mostrada en la Figura 25. Esta pregunta cuenta con opciones de tipo checkbox en vertical y una última respuesta abierta.

**INE Instituto Nacional de Estadística**

01 El INE 03 Métodos y estándares 05 Prensa 07 Censo electoral  
02 INEbase 04 Formación y empleo 06 Productos y servicios 08 Sede electrónica

Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE

Le quedan por responder como máximo 3 preguntas.

**5. Principales razones por las que el nivel de confianza es bajo.**  
Por favor, seleccione un máximo de 2 respuestas de las siguientes:

- ☒ No confío en los datos (por mi experiencia profesional)
- ☒ He oído/leído algo negativo sobre los datos
- ☐ Los datos son difíciles de entender
- ☐ Imagino interferencia política
- ☐ Otras respuestas: (indique cuáles)

[Anterior](#) [Siguiente](#)

© INE 2013 | [Accesibilidad](#) | [Aviso de seguridad](#) | [Aviso legal](#) | [Ayuda](#) | [Dónde encontramos](#)

[Año internacional de la Estadística](#) | [Sistema estadístico europeo](#) | [Contactar con INE](#)

*Figura 25. Quinta pregunta del cuestionario*

Tras esta pregunta se sigue el flujo normal hasta la última pregunta, que se muestra en la Figura 26. Esta pregunta tiene 6 opciones de tipo radio vertical con una última opción acompañada de un cuadro de texto para introducir una respuesta abierta.

**INE Instituto Nacional de Estadística**

01 El INE 03 Métodos y estándares 05 Prensa 07 Censo electoral  
02 INEbase 04 Formación y empleo 06 Productos y servicios 08 Sede electrónica

Encuesta: Información sobre el Sistema Estadístico Europeo en la web del INE

Le queda por responder como máximo 1 pregunta.

**6. ¿Cuál de los siguientes términos describe mejor su perfil?:**

- ☐ Profesor y/o estudiante
- ☐ Investigador, analista
- ☒ Responsable de políticas públicas
- ☐ Responsable de decisiones empresariales
- ☐ Medios de comunicación
- ☐ Otras respuestas: (indique cuáles)

[Anterior](#) [Siguiente](#)

© INE 2013 | [Accesibilidad](#) | [Aviso de seguridad](#) | [Aviso legal](#) | [Ayuda](#) | [Dónde encontramos](#)

[Año internacional de la Estadística](#) | [Sistema estadístico europeo](#) | [Contactar con INE](#)

*Figura 26. Última pregunta del cuestionario*



## **Anexo II. Soporte multi-idioma**

La herramienta desarrollada cuenta con un soporte multi-idioma gracias al cual la aplicación se puede visualizar en tantos idiomas como se quiera sin tener que crear tantas encuestas en base de datos como idiomas se quiera tener.

Esta aportación se consigue utilizando un paquete de idiomas que incluye tablas, triggers, procedimientos almacenados y funciones.

Para cada campo de texto del esquema al que se quiera dotar de servicio multi-idioma, se tiene una columna *FKI\_nombrecampo* que relaciona el campo con el identificador de una tabla llamada *TEXT\_TABLA*. Esta tabla contiene un identificador asignado a cada tabla con 1 o más campos a traducir. Dicho identificador y el nombre del campo (*nombrecampo*) se insertan en otra tabla llamada *TEXT\_TABLA\_CAMPO* donde se da otro identificador numérico al campo particular de la tabla.

El esquema de base de datos contiene una tabla llamada *TEXT\_VARCHAR* en la que se tiene un identificador del registro traducido, un identificador del idioma y el texto en dicho idioma. El identificador del registro traducido de la tabla *TEXT\_VARCHAR* coincide con el valor de *FKI\_nombrecampo* del registro. Un ejemplo de tablas que contienen campos sensibles a traducción se puede ver en la Figura 15 en las tablas PREGUNTA y ENCUESTA, cuyos campos *texto* y *presentación* van acompañados de sus correspondientes *fki\_texto* y *fki\_presentacion*.

Cada tabla que tenga uno o más campos con posibilidad de traducción, esto es que haya una columna de nombre *FKI\_nombrecampo*, debe contener un trigger en el que se contemplarán los casos de inserción, actualización y eliminación de dichos campos. En dicho trigger se trabajará con un identificador de idioma que será el predeterminado.

En las consultas realizadas para extraer preguntas y subpreguntas, los campos cuyos textos sean sensibles a traducción, deberán extraerse llamando a una función a la que se le pasará el idioma deseado y el identificador del campo a traducir. Esta función devolverá el texto contenido en la tabla *TEXT\_VARCHAR* cuyo identificador de campo traducido e idioma coincida con los recibidos como parámetros de entrada.

## **Anexo III. Accesibilidad**

El principal motivo para hacer que un sitio Web sea accesible es la total eliminación de barreras tecnológicas para cualquier persona. Tras finalizar la implementación de la parte gráfica de la aplicación se tienen que pasar las pruebas de accesibilidad necesarias de manera que la página Web cumpla con el nivel de conformidad “Doble A”, que implica que todos los puntos de verificación de prioridad 1 y 2 se satisfacen.

Se realizan pruebas de accesibilidad sobre las pautas marcadas por la W3C<sup>18</sup>, una comunidad internacional dedicada al desarrollo de estándares, en su documento WCAG 1.0<sup>19</sup>, nacido en mayo de 1999, y por la norma UNE 139803:2004<sup>20</sup>.

Se revisa que se cumplan las pautas y/o requisitos de prioridad 1 y 2, utilizando las herramientas PISTA<sup>21</sup> y TAW<sup>22</sup>. PISTA Accesibilidad es un proyecto promovido por el Ministerio de Industria, Turismo y Comercio con el objetivo de ofrecer a las Administraciones Públicas y otras empresas, una herramienta de análisis de las pautas sobre accesibilidad en los sitios Web. TAW es también una herramienta de accesibilidad de sitios Web que puede utilizarse tanto vía Web como por medio de una aplicación del ordenador.

Se analizan 19 requisitos de prioridad 1 entre ambas aplicaciones con un resultado satisfactorio pues no se muestra ningún error automático, los cuáles son importantes y tienen que ser corregidos. Los errores manuales que se obtienen son revisados y se consideran como advertencias ya que son puntos que no se pueden comprobar automáticamente.

Tras pasar las pruebas de verificación de prioridad 1 se verifican las pautas de prioridad 2 con las mismas herramientas. De nuevo no se obtiene ningún error automático en los 32 requisitos revisados, y en cuanto a errores manuales se obtiene una cantidad mínima y se revisan para comprobar si hay algún error grave.

Tras finalizar las pruebas de verificación de prioridad 1 y 2 se puede confirmar que la página Web cumple el nivel de conformidad “Doble A”.

---

<sup>18</sup> World Wide Web Consortium, <http://www.w3.org/Consortium/>

<sup>19</sup> Web Content Accessibility Guidelines 1.0, <http://www.w3.org/TR/WCAG10/>

<sup>20</sup> Aplicaciones informáticas para personas con discapacidad. Requisitos de accesibilidad para contenidos en la Web, [http://www.inteco.es/Accesibilidad/difusion/Normativa/Descarga/DescargaUNE\\_139803/](http://www.inteco.es/Accesibilidad/difusion/Normativa/Descarga/DescargaUNE_139803/)

<sup>21</sup> Pista Accesibilidad, <http://www.pistaaccesibilidad.com/>

<sup>22</sup> TAW, <http://www.tawdis.net/info/?lang=es>